



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV PROCESNÍHO INŽENÝRSTVÍ

INSTITUTE OF PROCESS ENGINEERING

PŘÍPRAVA NÁSTROJE PRO INŽENÝRSKÉ VÝPOČTY

DEVELOPMENT OF COMPUTATIONAL TOOL FOR ENGINEERING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jan Komiš

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Vondál, Ph.D.

BRNO 2019

Zadání bakalářské práce

Ústav: Ústav procesního inženýrství
Student: **Jan Komiš**
Studijní program: Strojírenství
Studijní obor: Základy strojního inženýrství
Vedoucí práce: **Ing. Jiří Vondál, Ph.D.**
Akademický rok: 2018/19

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Příprava nástroje pro inženýrské výpočty

Stručná charakteristika problematiky úkolu:

Bude zpracován postup pro snadnou instalaci výpočetních nástrojů pro základní inženýrské výpočty (vyprazdňování nádrže, výpočet výměníku tepla, vypařování, apod.). Připraveny a odzkoušeny budou výpočty vlastností materiálů (hustota, viskozita, měrná tepelná kapacita) pomocí volně dostupných nástrojů v moderním prostředí Python. Připraveny budou šablony pro tyto základní výpočty.

Cíle bakalářské práce:

Provedení průzkumu dostupných nástrojů pro inženýrské výpočty v Pythonu,
Příprava vhodného nastavení softwaru pro realizaci výpočtů,
Příprava minimálně pěti vzorových výpočtů pro základní úlohy z oblasti proudění,
Příprava ukázkového příkladu s využitím technologie Jupyter Notebook včetně interaktivních ovládacích prvků.

Seznam doporučené literatury:

Overview — Jupyter Documentation 4.1.1 alpha documentation [online]. [vid. 2017-10-27]. Dostupné z: <https://jupyter.readthedocs.io/en/latest/>

MCKINNEY, Wes. pandas: powerful Python data analysis toolkit [online]. 2017. Dostupné z: <https://pandas.pydata.org/pandas-docs/stable/pandas.pdf>

GOODWIN, David G. Working with Liquid/Vapor Fluid Models [online]. 2004. Dostupné z: <https://www.et.byu.edu/~tom/classes/641/Cantera/liquidvapor.pdf>

GOODWIN, David G. Defining Phases and Interfaces [online]. 2003. Dostupné z: <https://www.et.byu.edu/~tom/classes/641/Cantera/ inputFile/definingphases.pdf>

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

prof. Ing. Petr Stehlík, CSc., dr. h. c.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstract

Předmětem práce je seznámit čtenáře s použitím Pythonu v oblasti inženýrských výpočtů. Práce je rozdělena na teoretickou a praktickou část. Teoretická část seznamuje čtenáře s Pythonem a možnostmi jeho využití ve strojírenské oblasti. Druhá polovina teoretické části práce má za cíl seznámit čtenáře se základním výběrem studijních zdrojů a poskytuje návod, jak nastavit Python z pohledu běžného uživatele. Poslední část práce se zabývá výběrem vhodných knihoven (Scipy, Numpy, Cantera,...) a nastiňuje praktickou ukázkou použití na konkrétních úlohách z oblasti termomechaniky a hydromechaniky.

Klíčová slova:

Python, Jupyter, Cantera, Coolprop, Scipy, Pandas

Abstract

The thesis aims to provide insight into the use of Python in engineering calculations. The thesis has been divided into a theoretical and practical part. The theoretical part introduces Python and its possible uses in the field of engineering. The second section of the theoretical part describes key choices of study resources and provides instructions on how to set up Python from the perspective of a common user. The final part of the thesis deals with the choice of appropriate libraries (SciPy, NumPy, Cantera, etc.) and shows how these can be used practically in specific tasks from the fields of thermomechanics and hydromechanics.

Key words:

Python, Jupyter, Cantera, Coolprop, Scipy, Pandas

Bibliografická citace

KOMIŠ, Jan. Příprava nástroje pro inženýrské výpočty [online]. Brno, 2019 [cit. 2019-05-23]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/116680>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav procesního inženýrství. Vedoucí práce Jiří Vondál.

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci na téma Příprava nástroje pro inženýrské výpočty vypracoval samostatně, na základě konzultací s vedoucím bakalářské práce a s použitím odborné literatury.

V Brně dne 24.5.2019

.....
Jan Komíš

Poděkování

Tímto bych chtěl poděkovat vedoucímu mé práce panu Ing. Jiřímu Vondálovi, Ph.D., za cenné připomínky a rady týkající se mé bakalářské práce. Zároveň děkuji mé rodině za podporu při studiu.

Obsah

1. Úvod	1
2. Proč Python?.....	2
2.1. Vhodnost Pythonu pro inženýry.....	3
2.1.1. Univerzálnost jazyka Python	3
2.1.2. Dobře čitelný.....	3
2.1.3. Rychlost	4
2.1.4. Oblíbenost.....	4
2.1.5. Licence	5
2.2. Srovnání s MS Excel	6
3. Využití Pythonu	7
3.1. Použití ve výuce	7
3.1.1. George Washington University – MAE 6226 Aerodynamics– Hydrodynamics.....	7
3.1.2. University of California – Introducing Resonance	7
3.1.3. University of Notre Dame – CBE 20255 – Intro to Chemical Engineering ..	8
3.2. Použití Pythonu v praxi	8
3.2.1. Použití v aplikacích.....	8
3.2.2. Použití s hardwarem.....	10
4. Studijní zdroje.....	11
4.1. Literatura	11
4.2. MOOC.....	11
4.2.1. edX.....	11
4.2.2. Coursera	12
4.2.3. Udacity.....	12
4.3. Diskuzní portály	13
4.3.1. Stack Overflow	13
4.3.2. Google Groups	13
5. Instalace a správa Pythonu	14
5.1. Instalace Pythonu	14
5.2. Enviroments	16
5.3. Instalace knihoven.....	16
5.4. Vývojové prostředí.....	17
5.4.1. Spyder	18

5.4.2.	Jupyter Notebook	18
6.	Knihovny	20
6.1.	Knihovny pro matematické operace.....	20
6.1.1.	Numpy.....	20
6.1.2.	Scipy	21
6.2.	Knihovna pro vykreslování dat - Matplotlib	22
6.3.	Knihovna pro práci s daty - Pandas.....	24
6.4.	Knihovny pro GUI	25
6.4.1.	Jupyter Widgets	25
6.4.2.	PyQT	25
6.5.	Materiálové knihovny	26
6.5.1.	IAPWS	26
6.5.2.	CoolProp	27
6.5.3.	Cantera	28
6.6.	Knihovna pro práci se soubory.....	29
7.	Příklady.....	31
7.1.	Vypouštění vody z nádrže	31
7.1.1.	Zadání:	31
7.1.2.	Teoretický rozbor:.....	31
7.1.3.	Programové řešení:	34
7.1.4.	Kontrola výpočtu:	35
7.2.	Vypařování vody	35
7.2.1.	Zadání:	35
7.2.2.	Teorie:	35
7.2.3.	Programové řešení:	38
7.2.4.	Kontrola výpočtu:	38
7.3.	Tepelné ztráty zateplené stěny	39
7.3.1.	Zadání:	39
7.3.2.	Teorie:	40
7.3.3.	Programové řešení:	41
7.3.4.	Kontrola výpočtu:	42
7.4.	Částečné vypaření směsi	43
7.4.1.	Zadání	43
7.4.2.	Teorie	44
7.4.3.	Program:.....	45

7.4.4.	Kontrola výpočtu:	45
7.5.	Spalování plynné směsi.....	46
7.5.1.	Zadání	46
7.5.2.	Teoretický rozbor:.....	46
7.5.3.	Program:.....	51
7.5.4.	Kontrola výpočtu:	52
8.	Závěr.....	54
9.	Seznam použité literatury	55
10.	Seznam použitých zkratk a symbolů.....	62
11.	Seznam příloh	66

1. Úvod

S tím, jak roste výkon výpočetní techniky, roste i objem dat, která se už prakticky nedají zpracovávat ručně. Proto není divu, že vládní i soukromý sektor ve formě nadnárodních firem typu Amazon, Lockheed Martin, General Motors atd. investuje obrovské částky do vzdělání programování [1][2][3].

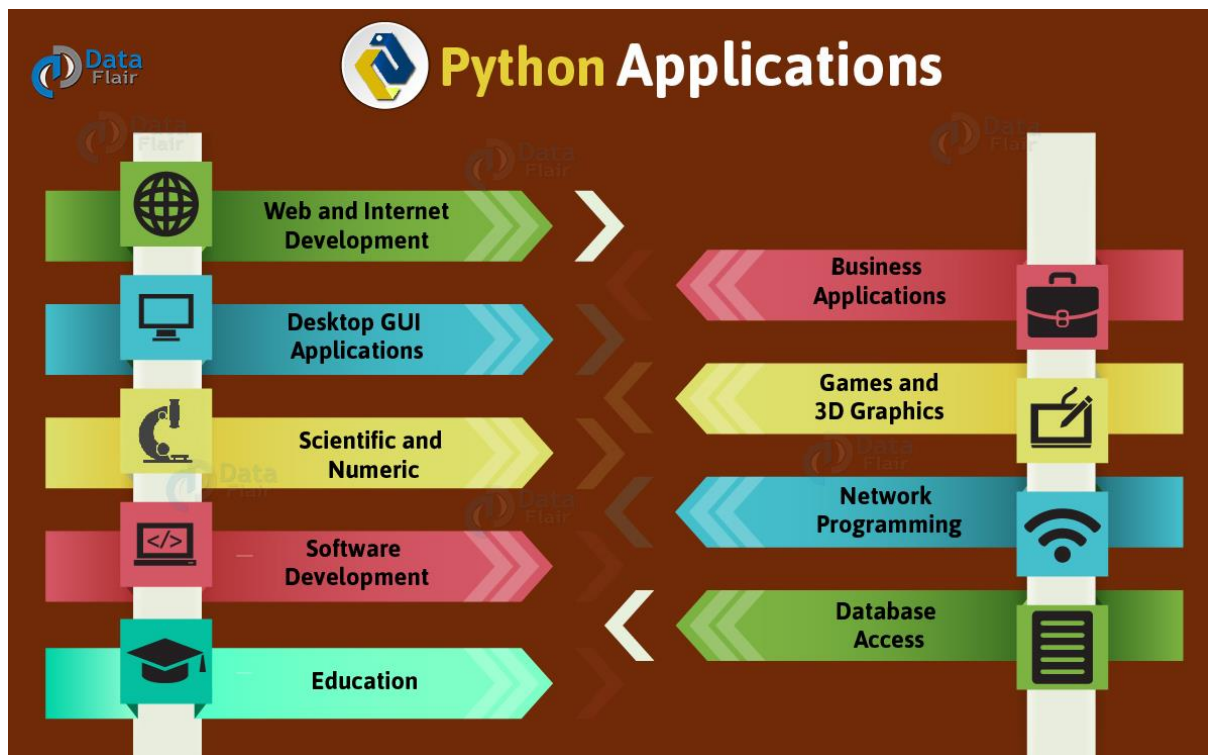
Tento fakt vzbuzuje otázku, jaký programovací jazyk je vhodný v inženýrské praxi. V dnešní době existuje nespočet programovacích jazyků – od jazyků, které byly vyvinuty v druhé polovině 20. století, až po jazyky, které jsou staré teprve pár let. Stejně jak se mezi sebou liší ve stáří, jsou velice rozdílné i v tom, na co se primárně hodí. V poslední době se velmi skloňuje použití Pythonu ve vědecké komunitě.

Python je dynamický interpretovaný jazyk, který zároveň podporuje různá programovací paradigmaty. Jeho tvůrcem je Guido van Rossum, který na něm začal pracovat na konci 80. let 20. století [4]. Název je odvozený od britské televizní show “Monty Python’s Flying Circus”. Jeho jednoduchá syntax byla z velké části odvozena od jazyka ABC, který sloužil v 80. letech pro výuku programování a pro vytváření prototypů. Na rozdíl od jazyka ABC, který se moc neujal, obliba Pythonu v posledních letech roste a často obsazuje horní místa v žebříčcích oblíbenosti [5].

Tato práce si klade za cíl nejenom objasnit popularitu Pythonu, ale i možné aplikace Pythonu pro účely inženýrských výpočtů.

2. Proč Python?

Python je široce používaný jazyk v mnoha oblastech – od tvorby webů přes strojové učení až po big data viz Obr. 2.1. Vzhledem k tématu této práce se bude diskuze zaměřovat pouze na relativně úzkou oblast využití programovacího jazyka pro oblast inženýrských výpočtů a potřeb technické praxe.



Obr. 2.1 Oblast použití Pythonu

- převzato z [6]

Python application – uplatnění Pythonu

Web and Internet Development - vývoj pro web

Desktop GUI Application – počítačové aplikace s GUI

Scientific and numeric – věda a numerické výpočty

Software development – vývoj softwaru

Education – výuka

Business Applications – podnikové aplikace

Games and 3D graphics – hry a 3d grafika

Network programming – síťové programování

Database access – přístup k databázím

2.1. Vhodnost Pythonu pro inženýry

Níže uvádím 5 nejčastějších důvodů, proč je Python oblíbený mezi vědeckou komunitou:

2.1.1. Univerzálnost jazyka Python

Jak bylo v úvodu kapitoly zmíněno, Python má široké spektrum využití. Tvorba webových stránek nebo správa databází není pravděpodobně to, co by inženýr využil ve své práci. Co ale může ocenit, jsou:

- Knihovny pro numerické výpočty

Python je velice hojně využíván vědeckou komunitou. Velkou zásluhu na tom má velké množství nejrůznějších vědeckých knihoven ať už pro numerické výpočty (NumPy, Scipy...), pro vizualizaci dat (Matplotlib, Bokeh...), nebo přímo pro specializované knihovny (zpracování obrazů – OpenCv, strojové učení – scikit-learn).

- API

Řada webových, ale i desktopových aplikací obsahuje API pro Python. Výraz API je dle encyklopedie Britannica [7] „*soubor standardizovaných požadavků, které umožňují různým počítačovým programům komunikovat mezi sebou*“. Prakticky to znamená, že lze např. celkem jednoduše napsat program pro průběžné odesílání dat díky Gmail API [8], nebo změnit geometrii v FreeCadu [9].

- Programování hardwaru

Python lze použít i pro programování hardwaru. Ať už se jedná o ovládání Arduina [10], Raspberry Pi [11], nebo ovládání vlastního zařízení postaveného na bázi mikrokontrolerů ovládaných knihovnou MicroPython [12].

2.1.2. Dobře čitelný

Jak už bylo uvedeno v předchozí kapitole, Python si zakládá na dobré čitelnosti kódu. Toho je docíleno hlavně dvěma způsoby.

První způsob by se dal pojmenovat jako přímočarost. Tím je myšleno, že spousta věcí se dá napsat velice intuitivním a krátkým způsobem. Ukažme si to na typickém příkladu. Prvním příkladem, kterým začíná každá programátorská učebnice a spočívá v zobrazení slov „Hello world!“. V Pythonu stačí napsat:

```
print ("Hello world!")
```

Kde v porovnání např. s Javou je nutné napsat:

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!")  
    }  
}
```

Identický příklad pro ostatní jazyky je lze nalézt na [13].

Druhý vliv na čitelnost kódu spočívá ve způsobu, jakým je kód odsazován. V každém programovacím jazyku je potřeba odlišit nějakým způsobem strukturu kódu. Častým způsobem je oddělení části kódu řešeno pomocí složených závorek spolu s odsazením (Java, C, C#). Python používá pouze odsazení pomocí tabulátoru nebo 4 mezer.

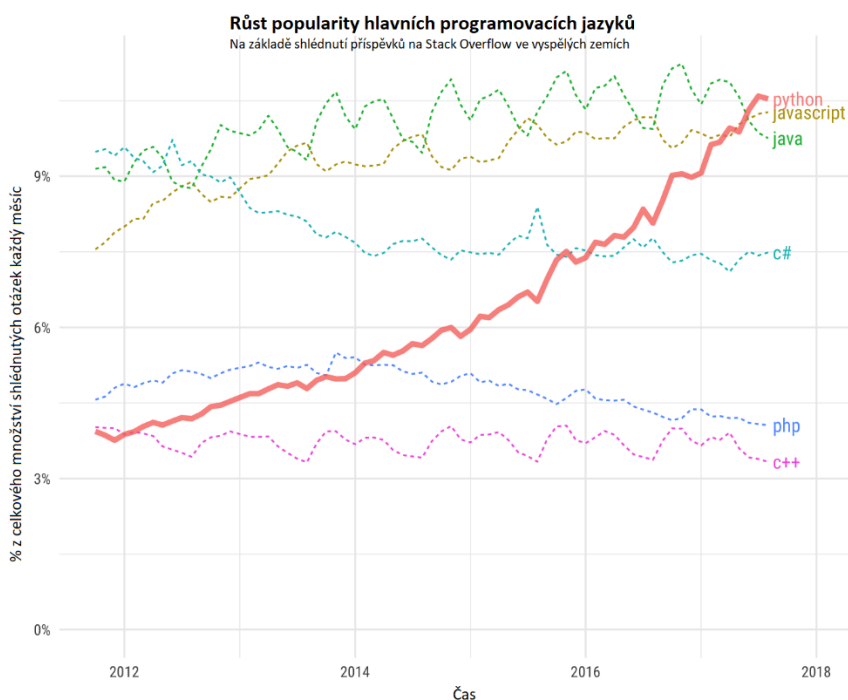
2.1.3. Rychlost

Jak bylo zmíněno v úvodu, Python je dynamicky interpretovaný jazyk. Interpretovaný jazyk potřebuje kromě zdrojového kódu i tzv. interpret (interpreter = tlumočnick), který tlumočí do strojového kódu. Dynamicky pak znamená jen to, že překládá pouze část kódu, který je zrovna nutný pro chod programu [14]. Pravým opakem jsou jazyky kompilované, které nejprve přeloží celý kód tzv. kompilátorem (compiler = překladatel) do strojového kódu a ten následně spustí. Typickým příkladem kompilovaného jazyka je C++[14].

Obecně panuje představa, že interpretované jazyky jsou automaticky pomalé. Toto tvrzení je pravdivé jen zčásti [15]. Nejsnadnější cestou, jak zrychlit program, je použít již nějakou existující knihovnu pro Python. Python má skvělou schopnost relativně snadno spolupracovat s jinými jazyky. Proto spousta knihoven není primárně psaná čistě v Pythonu, ale kombinuje i nějaký jiný jazyk – typicky C nebo C++. Případně lze využít některého alternativního interpretu pro Python, např. CPython. Srovnání rychlosti čistého Pythonu, C a některých Python knihoven (Scipy, Numpy,...) lze nalézt např. zde [16].

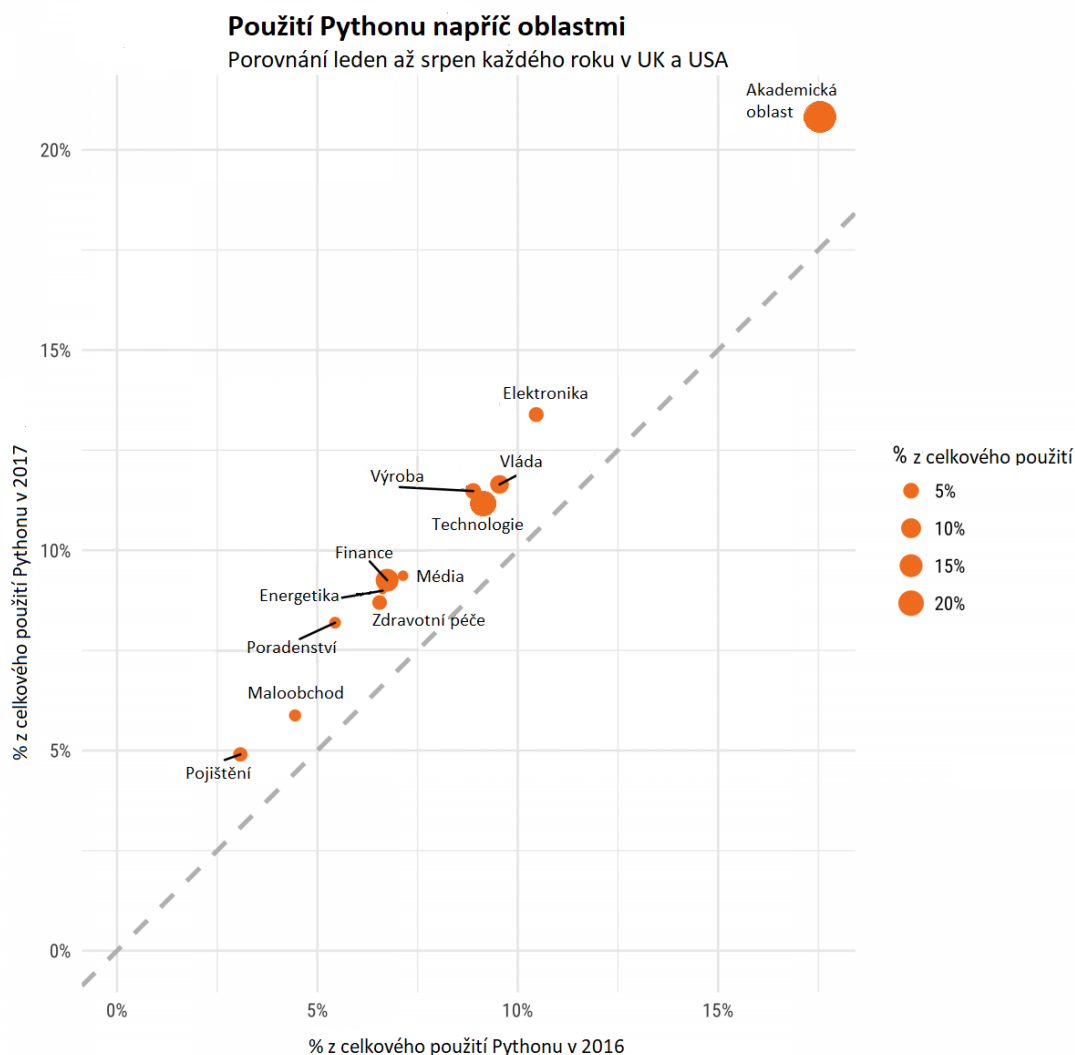
2.1.4. Oblíbenost

Python se poslední roky často objevuje na předních příčkách žebříčků nejoblíbenějších programovacích jazyků [17][18]. Tyto žebříčky jsou zpravidla kombinací několika faktorů, kde často figuruje skutečnost, jak moc se daný jazyk objevuje na pracovních portálech spolu s množstvím dotazů na Stack Overflow (viz kap. 4.3.1).



Obr. 2.2 Růst popularity Pythonu – převzato a přeloženo z [19]

Tyto průzkumy jsou ale v případě Pythonu velice zavádějící z důvodu velké různorodosti použití Pythonu. Z tohoto důvodu vznikl průzkum zaměřený na popularitu Pythonu napříč průmyslem také na portálu Stack Overflow [19]. Zde je vidět, jak Python dominuje v akademické sféře a je následován elektrotechnickým průmyslem.



Obr. 2.3 Popularita Pythonu napříč oblastmi – převzato a přeloženo z [20]

Díky vysoké popularitě vzniká uzavřená smyčka, kdy s rostoucím množstvím programátorů, kteří využívají Python, vzniká stále nový obsah ve formě nových knihoven, což se zase zpětně promítá do přílivu nových programátorů.

2.1.5. Licence

Python a jeho knihovny jsou často distribuovány pod některou z open-source licencí [21].

Zjednodušeně se dají tyto licence rozdělit na dvě skupiny:

V první skupině jsou licence GPL a LGPL. Ty se zaměřují na šíření kódu a nelze u nich změnit licence (nebo jen částečně). Prakticky to znamená, že pokud program využívá část kódu s GPL licencí, tak celý program musí být licencován pod GPL. LGPL se liší v tom, že nelze změnit licenci jen u části programu, který využívá LGPL licenci.

Druhá skupina, do které patří licence APACHE, PSFL a MIT, nemá prakticky žádná uživatelská omezení z pohledu změny licence nebo úpravy kódu. Licence nejčastěji po uživateli požadují uvedení copyrightu a tzv. disclaimer neboli zřeknutí se odpovědnosti [22].

V praxi to znamená, že při vývoji komerčního softwaru je nutné si pohlídat, jestli daná knihovna nepoužívá GPL licenci, a pokud ano, jestli nenabízí i placenou licenci pro vývoj komerčního softwaru (např. PyQT). V opačném případě nejsou na uživatele kladeny prakticky žádné nároky.

2.2. Srovnání s MS Excel

MS Excel je v dnešní době hlavním nástrojem pro zpracování souborů dat široké veřejnosti – od studentů po manažery v nadnárodních korporacích. Flexibilita použití Excelu je velice široká a jeho použití je ve většině případů nenahraditelné už jen z důvodu masového rozšíření kancelářského balíku Microsoft Office.

Je nesmysl porovnávat Excel a Python v možnostech využití. Pokud se ale zaměříme pouze na oblast zpracování dat, pro kterou je Excel primárně určen, lze nalézt několik případů, kdy je vhodnější využít Python nebo jej zkombinovat s Excelem (neuvažujeme pokročilé znalosti VBA):

- **Automatizace**

V situacích, kdy zpracováváme data opakovaně nebo je dat velké množství, je vhodnější použít program, než to dělat ručně. Pro tyto účely existuje řada knihoven pro automatizaci samotného Excelu (PyXXL, xlwings,...) až po kompletní zpracování dat (Pandas).

- **Sdílení dat**

Někdy má někdo jiný upravit naše data. V této situaci je rychlejší poslat program na úpravu dat, než vysvětlovat postup v Excelu.

- **Velké množství dat**

Práci s obrovskými soubory (několik stovek MB a víc) už Excel jednoduše nezvládá – ať už z důvodu využití limitu pro velikost excelového listu (1 048 576 řádků a 16 384 sloupců) [23], nebo jen proto, že není dostatek paměti RAM.

- **Import dat**

S Excelem se pojí velké množství chyb [24][25], které jsou způsobené importem nebo kopírováním dat. V drtivé většině případů to není problém Excelu, ale na vině byl lidský faktor. Python umožňuje import data z velkého množství zdrojů pomocí Pandas (CSV,SAS,xlsx,...) či celé řady dalších knihoven (html – BeautifulSoup, pdf – PyPDF2,...).

Neznamená to ale, že Excel je v porovnání s Pythonem horší. Excel a jiné tabulkové editory (Google sheets, OpenOffice Calc) jsou stále nenahraditelné pro zpracování malého množství dat a jejich vizualizace pomocí grafů. Ve výsledku je to o hledání kompromisu mezi tabulkovým editorem a programováním.

Porovnání Pythonu s jinými programovacími jazyky lze nalézt zde [26]

3. Využití Pythonu

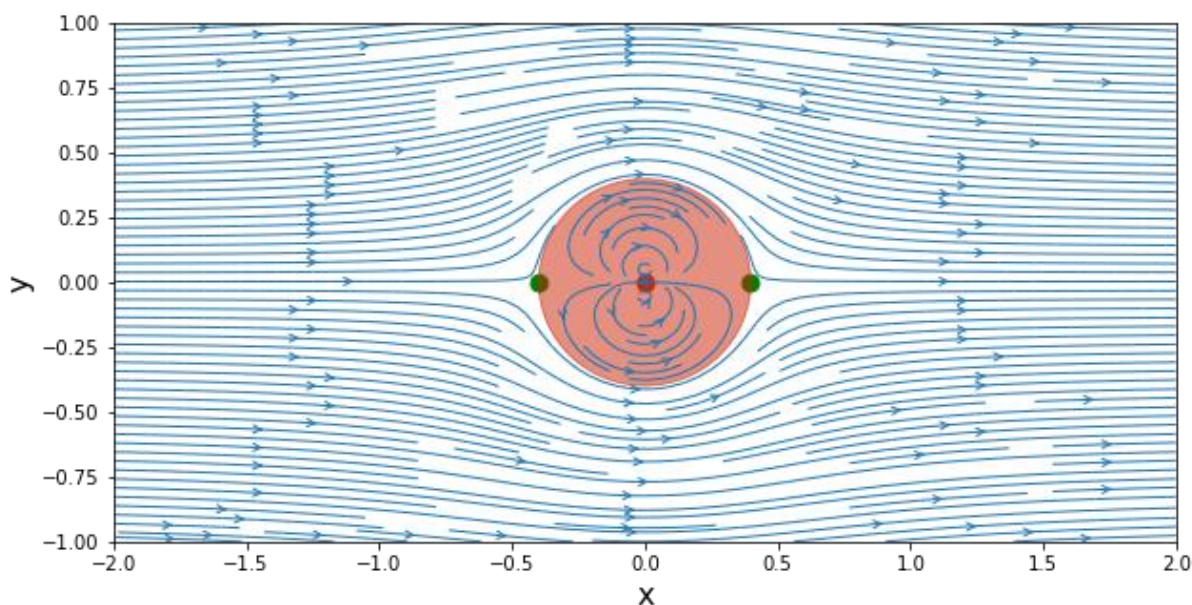
Jak bylo v úvodu druhé kapitoly uvedeno, tato práce se zaměřuje na použití pro inženýrské výpočty a potřeby technické praxe. Před výčtem použití si nejprve zodpovíme otázku, jestli a jak se používá Python v rámci univerzitní výuky.

3.1. Použití ve výuce

Python ve výuce se často využívá pouze pro vysvětlení základů programování na velké většině studijních oborů zaměřených na informatiku [27]. Na ostatních oborech ale lze Python nalézt i jako součást předmětů, které vyloženě nesouvisí s programováním.

3.1.1. George Washington University – MAE 6226 Aerodynamics–Hydrodynamics

Předmět zaměřený na základní teorie aerodynamiky (Navier-Stokesova rovnice, Helmholtzův teorém,...) aplikované v leteckém průmyslu [28]. Prekvizity předmětu předpokládají již znalost základů hydrodynamiky z předchozího předmětu APSC 6221: Mechanics of Fluids (lamilární a turbulentní proudění, odvození Navier-Stokesovy rovnice,...), matematické znalosti (derivování, integrování, diferenciální rovnice) a nějaké předchozí znalosti programování. Kurz používá výhradně Python, ale jeho znalost není přímo vyžadována. Počas kurzu se používají knihovny Numpy, Matplotlib a Scipy.



Obr. 3.1 Vizualizace toku kolem válce – převzato z [30]

Studijní materiály jsou ve formě Jupyter Notebooků, které jsou veřejně přístupné zde [29]. Kromě prezenčního studia, lze kurz absolvovat i online na MOOC univerzitním portálu [30].

3.1.2. University of California – Introducing Resonance

Volitelný předmět v osnovách Kalifornské univerzity. Účelem kurzu je seznámit studenty s vibračními mechanickými systémy, jež jsou popsány pomocí lineárních diferenciálních rovnic odvozených od Newtonova druhého zákona pohybu [31].

Výukové materiály jsou ve formě 14 Jupyter Notebooků, které nahrazují papírové studijní materiály. Kombinují Python kód s textem, matematickými rovnicemi, ilustracemi a YouTube videi. Všechny notebooky jsou veřejně dostupné na GitHubu.

Kurz využívá knihovny Matplotlib, Pandas, Numpy, Ipython Widgets, Scipy, SymPy a Resonance. Resonance je knihovna od autorů kurzu vytvořená za účelem usnadnění studentům modelování vibračních mechanismů bez velkého množství programování. Knihovna je dostupná zde [32] i se všemi výukovými materiály.

3.1.3. University of Notre Dame – CBE 20255 – Intro to Chemical Engineering

Prezenční kurz, jehož účelem je seznámit studenty s aplikací materiální a energetické rovnováhy pro řešení chemických procesů [33]. Kurz se točí okolo knihy *Introduction to Chemical Processes: Principles, Analysis, Synthesis*, kde některé příklady řeší studenti programováním. Studenti si mohou vybrat, jestli budou pracovat v Matlabu, na který má univerzita licenci, nebo v Pythonu. Výukové materiály s programy jsou přístupné zde [34].

Python materiály jsou v drtivé většině součástí notebooků spolu s příslušným doplňujícím textem nebo videi. Kurz nevyužívá žádnou materiálovou knihovnu a všechny materiálové vlastnosti jsou převzaty z literatury. Použité knihovny jsou pouze Scipy, Numpy, Matplotlib a SymPy.

3.2. Použití Pythonu v praxi

Mimo oblast výuky, kde Python hraje roli jako takové zpestření výuky daného předmětu, se teprve ukazují všechny možnosti použití. Největší pozornost se obecně dává všemožným knihovnám a tato práce není výjimkou (viz kap. 6). Existují ale i další dvě oblasti, kde Python má své místo:

3.2.1. Použití v aplikacích

Python lze v mnoha ohledech využít v řadě open-source, ale i u komerčních programů pro jejich automatizaci. Zde si ukážeme některé z nich.

FreeCad

FreeCad je open-source software (LGPL2+ license) pro parametrické 3D modelování. Kromě modelů pro strojírenství lze FreeCad využít i v ostatních oblastech, jako je např. architektura nebo elektrotechnika [9]. Ovládací prvky jsou podobné jako u komerčních CAD programů, jako je Inventor nebo Catia.

FreeCad obsahuje Python API díky kterému lze FreeCad ovládat přímo pomocí příkazového řádku. Díky Python skriptu lze např:

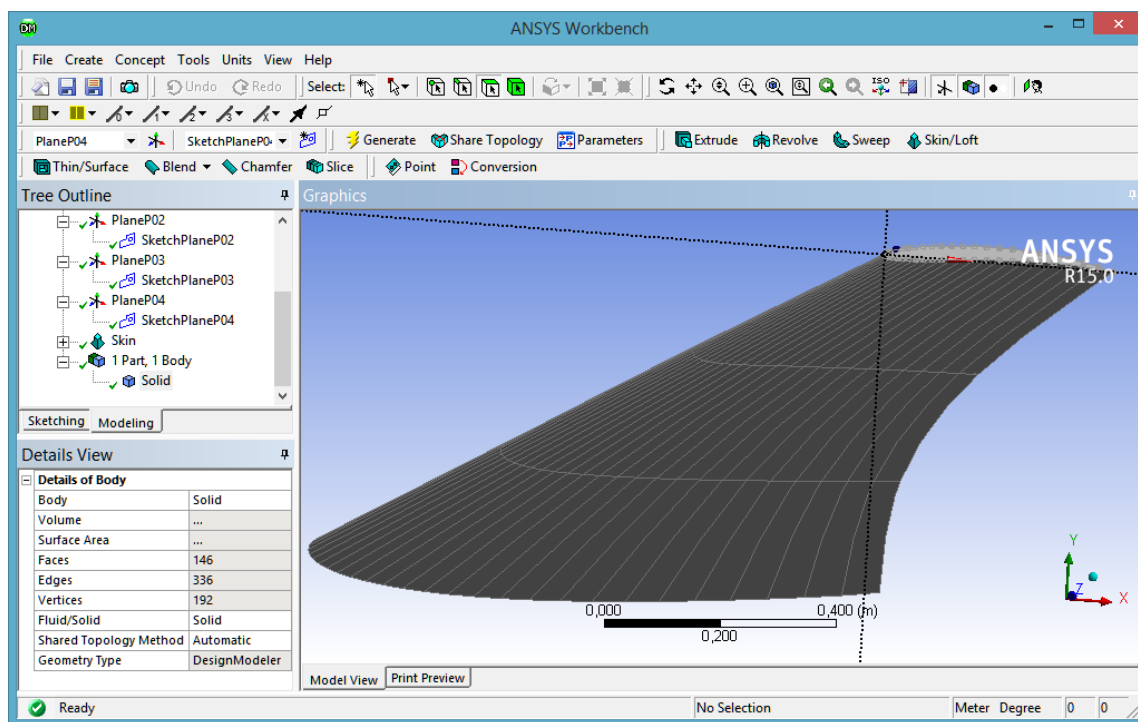
- vytvářet nové modely;
- upravovat stávající modely;
- měnit uživatelské prostředí FreeCadu;
- a další viz [35]

Ukázku změny rozměrů modelu pomocí skriptu lze vidět zde [36].

ANSYS

ANSYS od stejnojmenné společnosti je soubor programů určených pro širokou škálu inženýrských simulací [37]. Základním programem ANSYS je Workbench, který sdružuje ostatní simulační programy jako např. Fluent pro CFD simulace. Workbench používá skriptovací jazyk Python, ale jednotlivé prvky ANSYS jako již zmíněný Fluent používají JavaScript. Není ale přímo nutné znát JavaScript, jelikož programy umí zaznamenat sled operací a samy jej převedou do JavaScriptu. Následně se jen pospojují jednotlivé skripty pomocí Pythonu buď přímo v příkazové řádce, nebo také pomocí nahrávání sledu operací [38].

Ukázkou použití Pythonu je příklad uveřejněný na portále esss [39]. Zde použili Python jako mezičlánek pro propojení ANSYS Modelleru, který pomocí programu v JavaScriptu vytvořil geometrii křídla a následně Python tuto geometrii převedl do ANSYS Workbenche (viz Obr. 3.2).



Obr. 3.2 Vygenerovaná geometrie křídla – převzato z [44]

Abaqus

Abaqus je komerční program od společnosti 3DS pro tvorbu simulací metodou konečných prvků podobně jako ANSYS. Stejně jako ANSYS se skládá z více programů určených pro nejrozumnější inženýrské simulace (elektromagnetické pole, vibrace...) [40]. Na rozdíl od ANSYS Abaqus plně podporuje Python a celý proces preprocessingu (příprava geometrie – Abaqus/CAE) přes processing (test geometrie – Abaqus/Standard) až po postprocessing (vyhodnocení – Abaqus/CAE) [41].

Příkladem použití je článek [42], kde autor použil Abaqus a Python skript pro nalezení optimálních parametrů pro ohyb plechu.

3.2.2. Použití s hardwarem

S rozmachem uživatelsky přívětivého hardwaru si našel Python uplatnění i zde.

Arduino

Pod označením Arduino se skrývá otevřená platforma pro tvorbu elektronických obvodů. Skládá se z programovatelné obvodové desky (postavená na bázi Atmle AVR mikrokontrolérů) a vývojového prostředí (IDE – Integrated Development Environment) určeného pro psaní kódu. Programovacím jazykem Arduina je Wired, což je dle autorů jazyk vycházející z C++ [43]. Arduino ale lze programovat i pomocí Python, a to díky knihovně pySerial [44][10], která slouží jako kompilátor. Kód se vytváří v PC a je nutné jej do Arduina desky převést. Nejčastěji přes USB port.

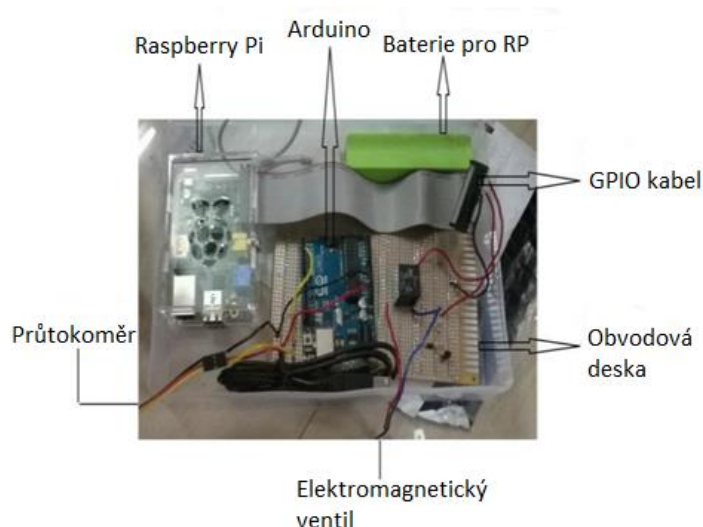
Pro Arduino existuje celá řada senzorů (snímač CO2, teplotní čidlo,...), které lze propojit s Arduinem pomocí GPIO pinu. Cena originálního Arduina se pohybuje okolo 500 Kč, ale díky platformě open-source lze nalézt celou řadu kopií v řádu desítek korun až stokorun.

Raspberry Pi

Raspberry Pi (dále jen RP) je malý jednočipový počítač, na kterém zpravidla běží některá verze Linuxu, ale lze zde instalovat i Windows [45]. Díky operačnímu systému pak není problém nainstalovat na RP Python. Obsahuje USB a HDMI port a případně další porty závislé na dané verzi. A podobně jako u Arduina lze připojit i další zařízení pomocí GPIO pinů. Cena RP se pohybuje okolo jednoho tisíce Kč.

Často se používá Arduino a RP zároveň, aby se využilo výhod obou platform. Silná stránka Arduina se díky kompilovanému kódu ukáže v případech, kdy je potřeba vykonávat jednoduchou činnost opakovaně (např. změř teplotu). Pro složitější využití (např. vykresli průběh teploty do grafu a odešli na mail) je zase vhodné využít RP.

Konkrétním příkladem použití je systém určený pro měření průtoku publikovaný zde [46]. Zde autor zkombinoval Arduino pro čtení signálu z průtokoměru a RP pro zpracování dat z Arduina a jejich následné odesílání díky napojení na webový server. RP zde zároveň kontroluje elektromagnetický ventil průtok.



Obr. 3.3 Zapojení kontroly průtoku - převzato a přeloženo z [48]

4. Studijní zdroje

Každá nová dovednost přináší řadu problémů, které je nutné překonat. Tento fakt je zvláště patrný při snaze naučit se programovat. Například Python je považován za velice přívětivý programovací jazyk z pohledu učení, ale to nic nemění na tom, že se nelze vyhnout desítkám, možná stovkám hodin studování nejrozumnějších informačních zdrojů. Proto jsou v této kapitole představeny nejrozumnější informační zdroje, které mohou pomoci se studiem.

4.1. Literatura

I v dnešní době, kdy má většina lidí díky internetu přístup prakticky k jakékoli informaci, stále vychází nepřehledné množství odborných knih o programování. Prakticky ke každé jen trochu známější Python knihovně existuje minimálně jedna kniha, kterou si můžete koupit třeba na Amazonu. U masově rozšířených knihoven typu Pandas lze nalézt klidně přes 30 knih.

To, že existuje velké množství knih, ještě nezaručuje jejich kvalitu. Nežádá-li se narazit na knihy, kde víc než polovina kódu je velice podobná kódu z originální dokumentace. Dalším problémem je rychlé zastarávání informací. To se týká zvláště úzce zaměřených knih na nějakou konkrétní knihovnu, která navíc v poslední době zaznamenala rapidní růst popularity. Jak moc se knihovna od vydání knihy změnila si lze zjistit v sekci *Release Notes*, která je součástí originální dokumentace knihovny.

V případě, že vás některá konkrétní kniha zaujala a je svým obsahem stále aktuální, není ve většině případů nutné ji hned kupovat a lze využít některou z možností výpůjčky. Kromě sítě knihoven v ČR lze využít možnost knihovny komunity kolem portálu python.cz [47] nebo si vybrat z řady ebooků, které zpřístupnila nadace Free Ebook Foundation [48].

4.2. MOOC

MOOC je anglická zkratka pro Massive Open Online Course. Význam této zkratky podle oxfordského slovníku [49] je to „*Kurz, který je dostupný přes internet, zdarma a velkému počtu lidí*“. Toto vyjádření je velice obecné a je způsobeno tím, že není nikde jasně dáno, co je a co není MOOC. Jediné, co existuje, je řada bodů, které by MOOC kurzy měly splňovat a podle kterých si zasloužily tuto zkratku [50]:

V dnešní době existuje mnoho poskytovatelů MOOC vzdělání. Níže jsou představeni tři nejznámější poskytovatelé:

4.2.1. edX

Výuková platforma založená roku 2012 ve spolupráci s univerzitami Harvard a MIT (Massachusetts Institute of Technology) [51]. K začátku roku 2019 obsahuje přes 1 900 kurzů z velké škály oborů, které jsou převážně v anglickém jazyce nebo s anglickými titulky. Lektori, kteří kurz organizují a vedou, jsou lektori ze světových univerzit (Berkeley, MIT, Harvard,...), v některých případech s podporou významné firmy v daném odvětví (Microsoft, ANSYS,...). Výuka probíhá ve formě řešení zadaných úkolů a testů, na které jsou studenti připravováni pomocí video přednášek, dodatečných studijních materiálů a související literatury. Každý kurz má také své diskuzní fórum, kde spolu studenti a lektori mohou komunikovat.

Dle průběhu uveřejňování obsahu se kurzy dělí na tzv. instructor-paced courses a na self-paced courses [51]. Instructor-paced jsou kurzy, které mají jasně daný začátek a konec kurzu a obsah spolu s kontrolními testy je vydáván postupně v průběhu kurzu. V dnešní době ale už na edX mnoho instructor-paced kurzů nenajdete a převažují self-paced kurzy. Ty mají jasně daný pouze

začátek, veškerý obsah je uveřejněný na začátku kurzu a je jedno, jak rychle budou studenti postupovat, protože neexistuje žádný rozvrh. Na první pohled se to může zdát jako výhoda, ale každý self-paced kurz má od svého tvůrce jasně danou délku trvání a po jejím uplynutí už nemusí být dostupná podpora jak ze strany autora kurzu, tak i ze strany studentů.

Většina kurzů je zdarma a je jen na studentech, jestli si zaplatí za certifikáty. Certifikáty slouží jako potvrzení o úspěšném absolvování kurzu nebo programu. Programy jsou série kurzů, které lze často absolvovat jednotlivě, ale není možné získat certifikát. Například u série programů MicroMasters je možné, že si studenti mohou požádat univerzitu, která daný program zaštituje, aby jim uznala kredity za obsahově stejné předměty. Jestli uznají absolvovaný program i jiné univerzity, je individuální záležitostí. Kromě toho jsou tyto kurzy velice také velmi žádané i mezi zaměstnavateli. Proto je dobré si na začátku zjistit, jestli člověk potřebuje po absolvování certifikát, protože certifikát nelze získat dodatečně po skončení kurzu.

4.2.2. Coursera

Výuková platforma založená roku 2012 profesory ze Standfordu. V dnešní době (začátek roku 2019) Coursera obsahuje kurzy od 149 univerzit a několika nadnárodních firem (Google, IBM,...), které vytvořily přes 2 000 kurzů rozličných specializací[52].

Coursera podobně jako jiné MOOC rozděluje výukový obsah do několika kategorií, podle množství obsahu a časové náročnosti. Základní distribucí výukového obsahu jsou kurzy. Styl výuky v kurzech je velice podobný jako u jiných MOOC platform. Oproti např. edX většina kurzů začíná v přesně daný termín a trvá zpravidla 4 až 6 týdnů. Takzvaných self-paced kurzů je na Courserě minimum. Podobně jako u edX si lze zaplatit certifikát dokládající absolvování kurzu.

Další kategorií studijního obsahu jsou tzv. specializace. Jejich specifikace jsou podobné jako programy u platformy edX. Předpokládaná délka u specializací je 4 až 6 měsíců. U specializací se platí ve formě předplatného. Ceny se pohybují v rozmezí 39 až 79 \$ měsíčně.

Poslední kategorií studijního obsahu jsou online tituly. Jde o speciální kategorii vzdělání, která svým obsahem a délkou trvání (1 až 4 roky) nahrazuje prezenční studium na dané univerzitě. Tomu ale také odpovídá cena studia, která se pohybuje v rozmezí 15 000 až 25 000 \$. Tento obsah už ale silně porušuje jeden ze znaků MOOC, a to je open. Kromě požadavků na předchozí zkušenosti je požadován vysokoškolský titul. Ve většině případů je nutné mít bakalářský titul nebo ekvivalentní typ vysokoškolského titulu.

4.2.3. Udacity

Udacity je výuková platforma, kterou založil v roce 2012 profesor ze Stanfordovy univerzity Sebastian Thrun společně s Peterem Norvigem [53]. Udacity se zpočátku zaměřovalo na univerzitní kurzy, ale v dnešní době se profiluje jako platforma nabízející kurzy vedené odborníky na danou oblast. Kurzy, na rozdíl od předchozích dvou platform, jsou zaměřeny kromě pár výjimek (statistika, úvod do fyziky,...) především na aplikaci IT technologií. Kromě typických IT témat (webdesign, machine learning,...) můžete absolvovat i netradiční kurzy – např. kurzy týkající se samořídících automobilů.

Výukový materiál je ve formě tzv. nanodegrees nebo samostatných kurzů. Samostatné kurzy jsou zdarma a slouží ve většině případů jako doporučený obsah před studiem některého z nanodegree programů, což jsou série kurzů podobného formátu jako u konkurenčních

platformem. Velkou devízou, jak již bylo zmíněno, je zpětná reakce od odborníků přímo z dané firmy, která na kurzu spolupracuje (Google, Uber,...).

Na Udacity převažují tzv. instructor-paced kurzy, u nichž platba probíhá jednorázově. Částky za nanodegree se pohybují okolo 1 000 \$. U self-paced kurzů někdy naleznete kromě jednorázové platby i měsíční předplatné, které se pohybuje okolo 100 \$. Po zaplacení Udacity umožňuje vrácení peněz do 7 dnů v případě, kdy vám z jakéhokoli důvodu kurz nevyhovuje.

Toto byly jen tři nejznámější MOOC platformy. Od roku 2012, kdy se začaly objevovat první platformy MOOC, uběhla už dlouhá doba, dnes (2019) existuje podle různých seznamů [54][55] něco kolem 40 poskytovatelů. Jak už ale bylo řečeno, není přesně dáno, co pod tento typ vzdělání zařadit, a proto se některé seznamy výrazně liší v počtu existujících MOOC platform [56].

S rostoucím počtem různých distributorů MOOC a tvůrců obsahu – ať už ze strany univerzit, nebo firem – roste množství nejrůznějších kurzů. Z tohoto důvodu vzniklo několik vyhledávačů, které vám usnadní výběr napříč MOOC distributory [57][58].

4.3. Diskuzní portály

4.3.1. Stack Overflow

Stack Overflow je diskuzní portál o všem, co souvisí s programováním. Tento portál vytvořila dvojice Jeff Atwood a Joel Spolsky v roce 2008 [19]. Jejich záměr dobře popsal Jeff Atwood na svém twitterovém účtu, kde v překladu napsal:

Přeji si, aby si více lidí uvědomilo, že cílem Stack Overflow není „odpověď na mou otázku“, ale „budeme spolupracovat na vybudování artefaktu, který pomůže budoucímu kodérovi“. [59]

Z tohoto důvodu nefunguje jako klasické diskuzní fórum, ale obsahuje i prvky známé z Wikipedie jako např. editace cizích příspěvků, přidávání hodnocení k jednotlivým odpovědím atd. To pro pasivní uživatele má pozitivní efekt v tom, že s největší pravděpodobností najdou spolehlivou odpověď na problém, který potřebují vyřešit. V dnešní době na SO je přes 17 miliónů zodpovězených otázek [19].

Bez registrace na SO lze číst všechny otázky a anonymně navrhnout editaci příspěvků. S registrací lze navíc pokládat otázky či komentovat stávající dotazy. SO má poměrně striktní pravidla pro dotazy/odpovědi. S aktivitou na SO je spojena i tzv. reputace. Tu lze získat především tvorbou kvalitních dotazů/odpovědí. S rostoucí reputací se zpřístupňují moderátorské nástroje.

4.3.2. Google Groups

Řešení problému, který se týká Pythonu nebo některé z oblíbených knihoven typu Pandas, Numpy atd., lze nalézt v řadě případů nejrychleji na SO. Na SO ale nelze najít odpovědi na méně známé knihovny – např. Cantera. V takovém případě je nejlepším způsobem na stránkách knihovny zjistit, jak se lze spojit s ostatními uživateli. Nejčastěji lze narazit na Google Groups.

Google Groups (dále jen GS) je bezplatná služba od Googlu, která umožňuje lidem diskutovat nad společným tématem [60]. Ať už se jedná o diskuzi nad firemním projektem nebo mezi spolužáky ve škole. Pro čtení příspěvku v GS není potřeba registrace. Ta je nutná jen v případě psaní dotazů. Pro registraci je ale nutné mít založený Google účet.

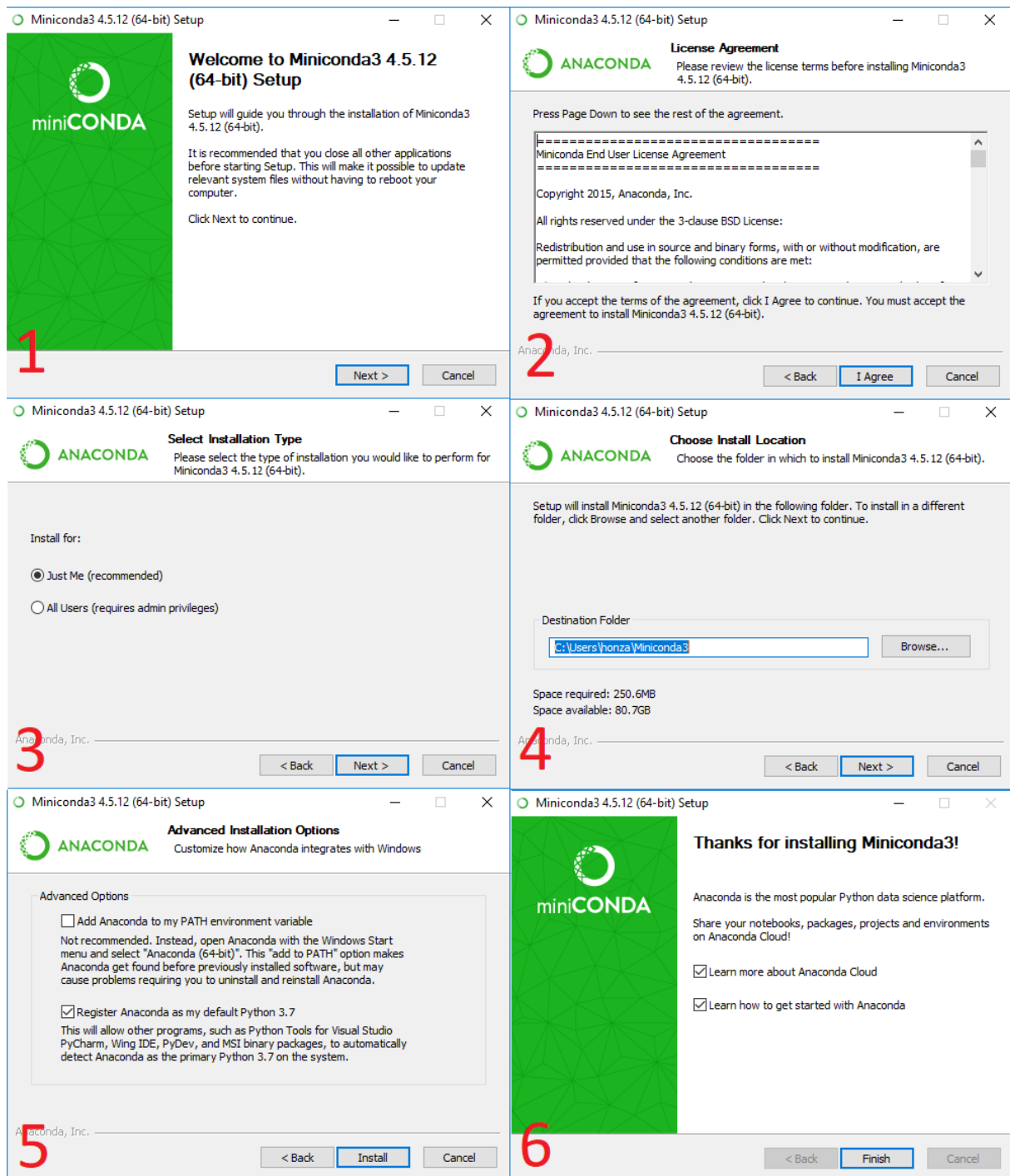
5. Instalace a správa Pythonu

Současný uživatel je zvyklý, že všechny software, a je jedno, jestli se jedná o ovladač k tiskárně nebo kancelářský balík Microsoft Office, se dá nainstalovat díky přiloženému instalačnímu souboru a dál už nic nemusí řešit. V případě Pythonu to tak nefunguje a je potřeba udělat několik kroků navíc, aby vše fungovalo tak, jak si přejeme.

5.1. Instalace Pythonu

Existuje velké množství způsobů, jak nainstalovat Python. V této práci se nebudeme zabývat originální distribucí přímo ze stránek python.org. Důvodem pro toto rozhodnutí je, dle mého názoru, výrazně složitější správa nainstalovaných balíčků. Místo toho Python nainstalujeme pomocí Minicondy [61].

Miniconda je neoficiální distribuce Pythonu od firmy ContinuumIO, díky které kromě Pythonu nainstalujeme Condu, což je program na správu knihoven a vývojových prostředí (tzv. environments) [62]. Instalační soubor Minicondy stáhnete z [61]. Je potřeba si vybrat instalační soubor dle operačního systému a verzi Pythonu. Na výběr je Python verze 2.7. nebo zrovna aktuální verze 3.X. Zde je vhodné si vybrat 3.X verzi, protože u verze 2.7. končí podpora ze strany vývojářů v roce 2020 [63]. Po stažení instalačního souboru a spuštění probíhá klasická instalace (viz Obr. 5.1). Miniconda potřebuje zhruba 300 MB volného místa na disku. Po skončení instalace by v nabídce Start měla přibýt položka Anaconda Prompt, což je prakticky příkazový řádek, který nalezneme ve Windows s tím rozdílem, že nemusíme zadávat adresář, kde jsme instalovali Minicondu, abychom mohli spouštět příkazy.



Obr. 5.1 Instalace Minicondy

5.2. Enviroments

Někdy se stane, že potřebujete spustit program, který používá staré verze balíčků nebo je napsaný v jiné verzi Pythonu (2. vs. 3. verze Pythonu). Není nutné hledat staré instalační soubory, ale lze to vyřešit mnohem jednodušší cestou, a to pomocí environments neboli prostředí[61].

Pro potřeby této bakalářské práce si vytvoříme vlastní prostředí pojmenované jako BP_thesis s verzí Pythonu 3.6.7, které vytvoříme dle následujících kroků:

1. Spustíme příkazový řádek Anaconda Prompt.

2. Zadáme příkaz:

```
conda create -n BP_thesis python=3.6.7
```

3. Až se objeví v příkazovém řádku věta.

```
Proceed ([y]/n)?
```

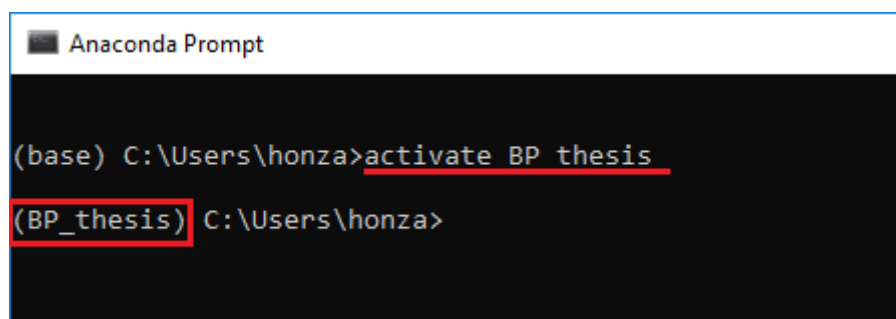
4. Napíšeme na klávesnici y a potvrdíme zmáčknutím enter.

Tímto je prostředí vytvořené a aktivujeme ho napsáním příkazu (viz Obr. 5.2):

```
activate BP_thesis
```

a deaktivujeme příkazem:

```
deactivate
```



Obr. 5.2 Aktivní prostředí

5.3. Instalace knihoven

Pokud chceme používat externí knihovny (např. Pandas) je nutné je nejprve doinstalovat. Podobně jako u instalace Pythonu i zde existuje celá řada možností, jak to provést. Zde si pro přehlednost ukážeme pouze jeden postup.

Před samotnou instalací je nutné nejprve najít instalační odkaz (v ang. dokumentaci pod pojmem channel) na danou knihovnu buď na stránkách [64] (viz Obr. 5.3), nebo jednoduše napsat do internetového vyhledávače „conda“ s názvem hledané knihovny.

Installers

conda install ?

linux-ppc64le	v0.24.2
osx-32	v0.16.2
linux-64	v0.24.2
win-32	v0.24.2
osx-64	v0.24.2
linux-32	v0.23.4
win-64	v0.24.2

To install this package with conda run:

```
conda install -c anaconda pandas
```

Obr. 5.3 Instalační odkaz knihovny

Alternativním způsobem jsou originální stránky autora knihovny, ale zde ve většině případů nalezneme pouze instalaci pomocí pipu, což je originální program pro správu knihoven od Pythonu.

Před instalací si přepneme prostředí na BP_thesis, až poté zkopírujeme odkaz do příkazového řádku. Protože v této práci je použita Pandas ve verzi 0.23.4, tak ještě dopíšeme za odkaz číslo verze:

```
conda install -c anaconda pandas=0.23.4
```

A potvrdíme enterem. Stejně jako v předchozí kapitole se objeví příkaz:

```
Proceed ([y]/n)?
```

Zmáčkne y a znovu potvrdíme.

Po skončení instalace pokud zadáme příkaz:

```
conda list
```

tak zobrazíme seznam všech nainstalovaných knihoven. Všechny ostatní příkazy spojené s knihovnami lze nalézt zde [61].

Protože všechny knihovny, které jsou v této práci použity, se instalují stejným způsobem, tak zde nejsou uvedeny. Jejich seznam je v příloze č.1 spolu s instalačními odkazy.

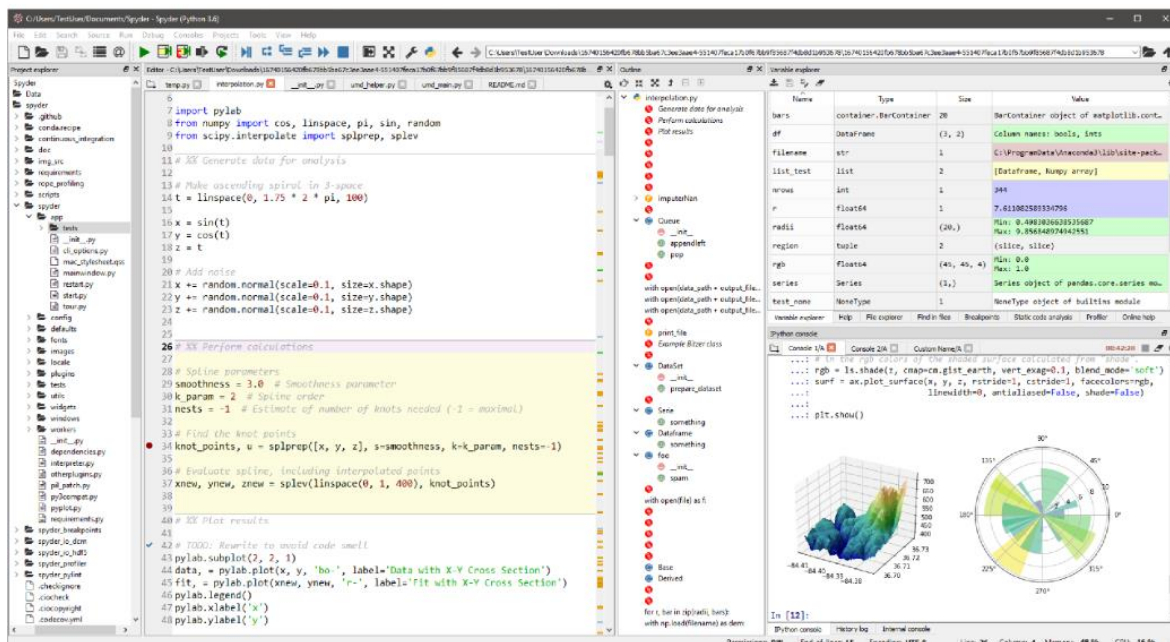
5.4. Vývojové prostředí

Po instalaci Pythonu a balíčků se dostáváme k neméně důležité části, a to volbě programovacího prostředí. Programovacích prostředí, která podporují Python, je celá řada. Na psaní kódu lze použít i obyčejný Poznámkový blok, který je součástí Windows. Práce programátora ale z větší části spočívá v testování a ladění kódu než v samotném psaní. Proto je vhodné využít Integrované Vývojové Prostředí (IDE). Jedná se o program, který má usnadnit psaní kódu. Kromě textového editoru běžně obsahuje konzoli pro spouštění kódu, debugger a další funkce, které zlepšují produktivitu práce.

Protože valná většina IDE se vzájemně moc neliší, zaměříme se pouze na dvě IDE, která jsou mezi vědeckou komunitou hodně oblíbená, a to Spyder a Jupyter.

5.4.1. Spyder

Spyder neboli Scientific Python Development Environment je multiplatformní IDE určené pro Python [65]. Spyder nepatří mezi nejznámější vývojová prostředí, jako je např. Visual Studio nebo Eclipse, ale mezi vědeckou komunitou je velice oblíben díky své provázanosti s knihovnami jako je Scipy, Matplotlib nebo Numpy.



Obr. 5.4 Prostředí Spyderu – převzato z [65]

Hlavní funkce Spyderu [65]:

- textový editor se zvýrazněním syntaxe;
- zobrazení všech proměnných;
- konzole pro spouštění kódu;
- debugger pro ladění kódu řádek po řádku;
- nástroj Profiler umožňuje zobrazit dobu běhu jednotlivých funkcí v kódu;
- integrovaná nápověda (i pro knihovny Scipy, Matplotlib,...);
- instalovat pluginy třetích stran

Instalační soubor lze stáhnout buď z [65], nebo nainstalovat pomocí příkazu:

```
conda install -c anaconda spyder
```

Popis základního nastavení viz originální dokumentace [66].

5.4.2. Jupyter Notebook

Jupyter Notebook (dále jen JN) je multiplatformní webové interaktivní prostředí pro tvorbu Notebook dokumentů. To kombinuje jak samotný kód (Python, Julia, R), který lze přímo spustit v notebooku, tak dodatečný obsah ať už se jedná o text, ilustraci či video. Tímto přístupem se

JN dost odlišuje od klasických IDE (např. Spyder, ATOM, Eclipse,...), blíže má k programům jako je Wolfram Mathematica [67] nebo MathCad [68].

Po stránce funkcí JN obsahují prakticky jen základ (zvýraznění syntaxe, nápověda, debugger). V případě dalších funkcionalit ale lze doinstalovat pluginy třetích stran, např. správce proměnných [69]. Na rozdíl od jiných IDE JN se efektivně využívá i jako prezentační nástroj, a proto obsahuje i možnost uložit celý notebook do PDF či html.

Instalace JN lze podobně jako u Spyderu instalovat buď přímo z webu JN [70], nebo pomocí příkazu:

```
conda install -c anaconda jupyter
```

Nastavení a práce v JN viz originální dokumentace [70].

Výběr vhodného IDE je hodně individuální záležitostí a neexistuje univerzální odpověď, které je nejlepší. Proto autor práce doporučuje nainstalovat obě jak Spyder tak i Jupyter (minimálně Jupyter Notebook pro spuštění příkladů v kap.7) a osobně zjistit, které více vyhovuje. Případně zvolit jiné na základě referencí z mnoha internetových zdrojů např. [71][72].

6. Knihovny

Python už v základu umí pracovat se stěžejními matematickými operacemi, jako jsou logaritmy nebo goniometrické funkce (modul *math*). Spoustu dodatečných funkcí (vykreslování grafů, integrální počet,...) je ale nutné zvlášť doinstalovat formou externích knihoven. Každá knihovna má jiného autora, což ve většině případů souvisí s rozličnou kvalitou dokumentace. Proto v této kapitole kromě představení, co daná knihovna umí a jakým způsobem ji využít, uvádí autor práce i doporučené studijní zdroje.

6.1. Knihovny pro matematické operace

Jak bylo zmíněno v úvodu kapitoly, Python umí řešit základní matematické operace. Pro složitější matematiku (např. maticový počet či integrace) je nutné doinstalovat příslušnou knihovnu.

6.1.1. Numpy

Funkce: Numpy [73] je základní balíček umožňující práci s vícerozměrnými maticemi. Kromě toho obsahuje i jiné nástroje – lineární algebru nebo Fourierovu transformaci.

Licence: BSD

Použití: Python sice už v základu umí pracovat s n-ticemi (v dokumentaci pod pojmem *listy*), ale neumí pracovat se všemi prvky n-ticemi zároveň a je nutné si často pomáhat *for cyklem*. Proto bylo Numpy použito jako náhrada za datový typ *list* v Pythonu a pro řešení soustavy rovnic.

Názorný příklad, na kterém lze ukázat elegantnější použití Numpy je v situaci, kdy je potřeba zvětšit prvky matice a o 1:

$$a = \begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix} \rightarrow \begin{pmatrix} 3 & 5 \\ 7 & 9 \end{pmatrix}$$

Při použití samotného Pythonu je zápis následující:

```
a = [[2,4],[6,8]]
```

```
a = [[y+1 for y in x] for x in a]
```

Pomocí Numpy je postup následující:

```
import numpy
```

```
a = numpy.array([[2, 4], [6, 8]])
```

```
a = a + 1
```

Pro řešení soustavy rovnic autor použil funkci *linalg*. Ukázka řešení soustavy rovnic je v příkladu částečného vypaření směsí (kap. 7.4).

Studijní materiály: Oficiální dokumentace je celkem dobře napsaná s řadou příkladů a obsahuje srozumitelný tutoriál. Pro Numpy ale existuje i řada lépe napsaných/natočených tutoriálů viz např. [74]. Pro řešení specifických problémů autor doporučuje podívat se na Stack Overflow [19].

6.1.2. Scipy

Funkce: Scipy [75] je referenční knihovnou pro řešení vyšší matematiky. Celá knihovna je rozdělena do několika modulů dle zaměření:

- **cluster** – algoritmy pro clusterové výpočty;
- **constants** – fyzikální a matematické konstanty;
- **fftpack** – Fourierova transformace;
- **integrate** – integrace a obyčejné diferenciální funkce;
- **interpolate** – interpolace;
- **io** – modul pro import a export dat z Matlabu;
- **linalg** – lineární algebra;
- **ndimage** – N-rozměrné zpracování obrazu;
- **odr** – ortogonální regrese;
- **optimize** – optimalizace a hledání kořenů;
- **signal** – zpracování signálů;
- **sparse** – řídké matice;
- **spatial** – prostorové datové struktury;
- **special** – speciální funkce;
- **stats** – statistika.

Licence: BSD

Použití: Z těchto modulů autor testoval, jaké je použití pro:

Derivaci

$$f(x) = \left(\frac{2x}{1-x^2} \right) \rightarrow f(x)' = \frac{2 \cdot (1+x^2)}{(1-x^2)^2} \rightarrow x = 3 \rightarrow = 0.3125$$

```
import scipy.misc
```

```
def f(x):  
    return (2*x)/(1-(x**2))
```

```
vysledek = scipy.misc.derivative(f, 3,dx=1e-6)  
print(vysledek)
```

Zobrazí výsledek 0.3125000000436806

Integraci

$$\int_0^1 \int_0^2 x^2 y e^{xy} dy dx = \dots = 2$$

```
import scipy.integrate  
from math import exp
```

```
def funkce(x,y):
    return (x**2)*y*(exp(x*y))

vysledek = scipy.integrate.dblquad(funkce, 0, 2, 0, 1)
```

Vypočte identickou hodnotu.

Obyčejnou diferenciální rovnici

$$y' = xy^2 \text{ pro } y(0) = 1 \rightarrow y = \frac{2}{(2-x^2)} \text{ pro } x = 0.5 \rightarrow = 1,1428$$

```
import scipy.integrate

def dy_dx(y,x):
    return x*(y**2)

xs= [0,0.5]
y0 = [1]
vysledek_y = scipy.integrate.odeint(dy_dx, y0, xs)
```

Kód zobrazí n-tici s hodnotami 1, 1.1428, které odpovídají $x = 0, 0.5$.

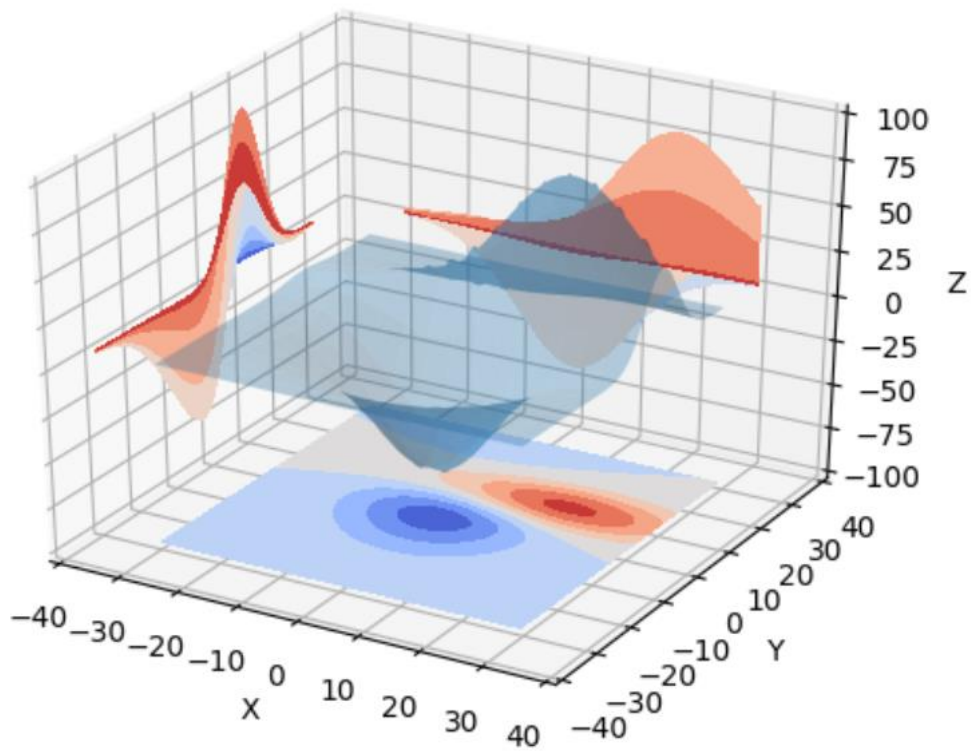
Studijní materiály: Velkou představu o knihovně lze získat přímo z oficiální dokumentace [75], která je celkem přehledná. Zároveň existuje i řada tutoriálů pro počáteční seznámení např. [76]. V případě problémů lze dobře využít Stack Overflow [19].

6.2. Knihovna pro vykreslování dat - Matplotlib

S výpočty souvisí i možnost vykreslení dat. Python neobsahuje žádnou funkci pro vykreslení grafů, a je proto nutné potřebnou knihovnu doinstalovat. Příkladem je nejrozšířenější knihovna Matplotlib [77]

Funkce: Matplotlib je knihovna pro vykreslování grafů. Od klasických čárových grafů, histogramů přes vektorová pole až po 3d grafy (viz obr.) Kromě toho umí:

- umožňuje vytvářet popisky v LATEXU;
- kvalitní grafický výstup, včetně formátů PNG, PDF, SVG, EPS;
- GUI pro interaktivní práci s grafy (zoomování, posouvání,...).



Obr. 6.1 3D graf s průmětem do rovin – převzato z[77]

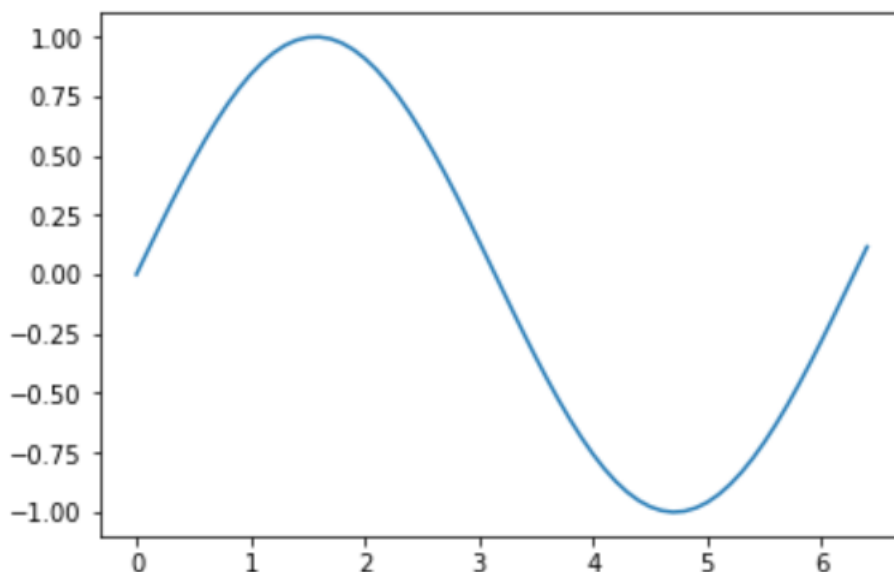
Licence: PSF (Python Software Foundation)

Použití: Přestože Matplotlib nabízí širokou škálu nejrozumnějších grafů a ještě větší množství následných úprav, nepromítá se to do nároků na uživatele. Matplotlib lze tedy jednoduše aplikovat i na jednoduché příklady, např. na vykreslení sinus funkce:

```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0, 6.5, 0.1)
y = np.sin(x)
fig, ax = plt.subplots()
ax.plot(x, y)
plt.show()
```

Vykreslený graf viz Obr. 6.2



Obr. 6.2 Sinus funkce v Matplotlib

Studijní zdroje: Originální dokumentace i přes svoji velikost je poměrně přehledná s řadou příkladů. Díky tomu, že Matplotlib je považována za referenční knihovnu, existuje i velká řada webových tutoriálů, např. [78]. Pro řešení specifických problémů doporučuji Stack Overflow.

6.3. Knihovna pro práci s daty - Pandas

V situacích, kdy je potřeba nějakým způsobem upravovat data (mazat duplicitní záznamy, filtrovat,...) je možnost využít Excel nebo použít specializovanou knihovnu. Příkladem je velmi rozšířená knihovna Pandas [79].

Funkce: Pandas je knihovna pro zpracování a analýzu dat, vyvinutá pouze pro Python. Funkce Pandas lze rozdělit do několika oblastí:

- datové struktury – vlastní datové struktury pro správu dat;
- indexovací nástroje – nástroje pro změnu indexů a třídění dat;
- transformace dat – nástroje na úpravu dat (smazání, přepisování,...);
- nástroje na kontrolu dat – hledání chybějících, duplicitních nebo jinak vadných dat;
- nástroje pro import/export dat – soubory txt, csv, xls, xlsx,...;
- statistika – funkce pro řešení statistických výpočtů (korelace, průměr,...).

Licence: BSD

Použití: Pomocí Pandas lze úplně nahradit zpracování dat např. v Excelu. Pomocí několika málo řádků kódu lze řešit filtraci dat, mazání duplicitních záznamů, ale i složitější věci typu extrahování čísla z textu či statistické funkce. Je však na zvážení, zda se to vyplatí z časového hlediska. Jediná funkce Pandas, která funguje objektivně lépe než v případě Excelu, je import dat. Import dat v Excelu není stoprocentní a někdy se stává, že např. špatně interpretuje časový formát.

Z toho důvodu by i v rámci malého množství dat autor práce doporučil použít Pandas pro extrakci dat a jejich následné uložení např. do Excelu.

Studijní materiály: Pandas má nejlépe napsanou dokumentaci [79], minimálně z knihoven uvedených v této kapitole. Na počáteční seznámení s knihovnou je dobré projít v sekci „Getting started“ kapitolu „10 Minutes to Pandas“, díky které se lze rychle zorientovat v možnostech knihovny. V sekci „User Guide“ autor doporučuje kapitolu „IO Tools (Text, CSV, HDF5,...)“ pro informace o tom, jak importovat data. Na specifické problémy je ideální najít řešení na Stack Overflow [19].

6.4. Knihovny pro GUI

Pro lepší uživatelský komfort je vhodné vytvořit k programu i grafické uživatelské prostředí neboli GUI (Graphical User Interface).

6.4.1. Jupyter Widgets

Funkce: Jupyter Widgets [80] nebo také pod starším názvem IPython Widgets je rozšíření do Jupyter Notebooku pro vytváření interaktivních prvků (tlačítka, posuvníky,...).

Licence: BSD

Použití: Tato knihovna je vhodná pouze pro lepší zážitek s Jupyter Notebookem, aby uživatel nemusel žádným způsobem zasahovat do kódu (např. změnit proměnné výpočtu) a měl lepší uživatelský zážitek. Ukázka použití je v kap. 7.1, kde je použito prakticky vše, co tato knihovna nabízí (různé druhy widgetů), propojení s kódem, formátování,...). Pro vytváření komplexních aplikací je vhodné využít specializovanou GUI knihovnu, např. PyQt či WxPython.

Studijní materiály: Originální dokumentace je celkem přehledná [80]. Na Stack Overflow v době psaní této práce prakticky nic není, ale existuje velmi aktivní skupina na Google Groups kolem Jupyter IDE, kde se dá ale probrat i Jupyter Widgets [81].

6.4.2. PyQt

Funkce: PyQt [82] je knihovna pro vytváření GUI, která je postavená na QT knihovně, což je multiplatformní knihovna primárně pro C++. QT knihovna kromě velké řady funkcí pro tvorbu GUI obsahuje nástroj program QT designer pro tvorbu GUI tzv. stylem drag and drop.

Licence: Duální licence GNU GPL v3 a licence společnosti Riverbank.

Použití: PyQt autor využil v kap. 7.2. Celé rozvržení programu je vytvořené v QT designeru a uložené jako soubor s koncovkou *.ui*. Ten následně přes příkazový řádek převedl do Python kódu příkazem:

```
pyuic5 nazev_souboru.ui -o nazev_souboru.py -x
```

Takto vzniklý kód už nijak neupravoval. Všechny funkce, které se staraly o chod výpočtu a propojovaly jednotlivé widgety, napsal zvlášť do dalšího souboru.

Tento postup autor doporučuje, protože někdy je potřeba upravit rozvržení prvků v GUI a stačí jen znovu otevřít *.ui* soubor v designeru, upravit např. nastavení velikosti písma a znovu převést do Python kódu.

Ruční programování GUI má výhodu v maximální kontrole nad kódem, dle autora je designer lepší z důvodu velké časové úspory ať už z pohledu dlouhého nastavování všech prvků GUI, tak i z pohledu menšího nároku na znalosti parametrů jednotlivých widgetů.

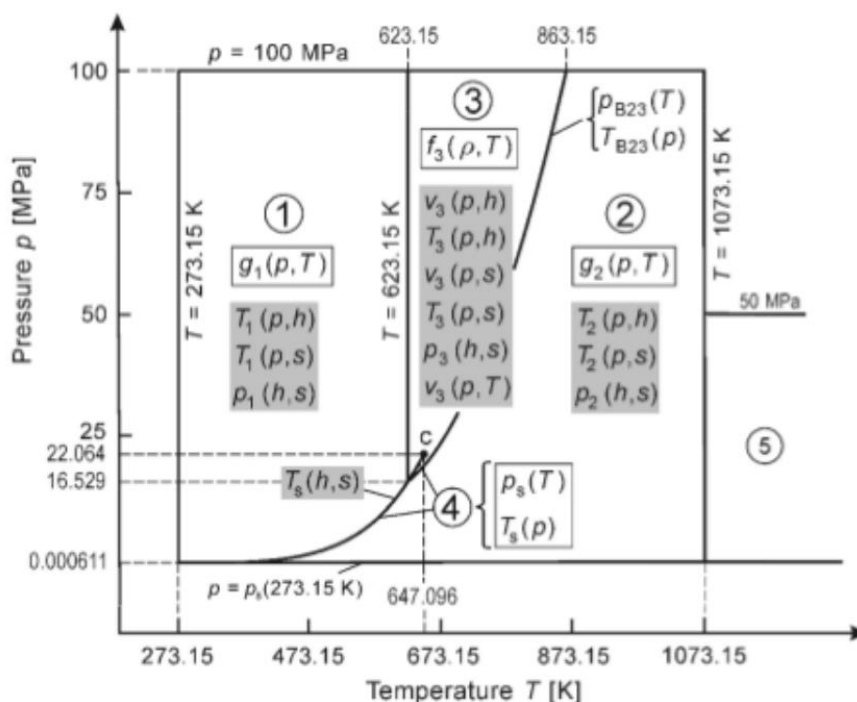
Studijní materiály: Originální dokumentace je obrovská a pro méně zkušeného uživatele je velice nepřehledná [82]. Pro počáteční seznámení autor doporučuje projít tento YouTube tutoriál [83] a pro seznámení s QT designerem pročíst originální dokumentaci [84]. Pro PyQT existuje i skvělá kniha [85]. Jde o vhodně strukturovaný tutoriál s řadou příkladů, ale jeho problémem je, že používá již starou verzi PyQT (PyQT4), a v případě použití je nutné si nastudovat rozdíly mezi PyQT4 a PyQT5. Při specifických problémech se lze spolehnout i na Stack Overflow[19], kde PyQT má velkou základnu uživatelů.

6.5. Materiálové knihovny

S inženýrskými výpočty souvisí i vhodný zdroj materiálových vlastností. Tato kapitola proto představí tři referenční knihovny s vlastnostmi plyných a kapalných látek.

6.5.1. IAPWS

- Funkce: IAPWS [86], jak už název napovídá, jsou importované tabulky mezinárodní asociace pro vlastnosti vody a vodní páry. Knihovna obsahuje tyto moduly:
- Iapws – obecné fyzikální vlastnosti vody (konduktivita, viskozita,...);
 - Utils – dodatečné nástroje (jaké je skupenství za daných podmínek,...);
 - Iapws95 – termodynamické vlastnosti vody dle vědecké směrnice IAPWS 95;
 - Iapws97 – termodynamické vlastnosti vody dle průmyslové směrnice IAPWS 97;
 - Iapws08 – fyzikální vlastnosti mořské vody dle směrnice IAPWS 08;
 - Ammonia – termodynamické vlastnosti směsi vody a amoniaku;
 - HumidAir – termodynamické vlastnosti vlhkého vzduchu.
- Licence: GPL-3
- Použití: Z knihovny autor testoval pouze modul *IAPWS97*. Ten je rozdělen do 5 regionů viz Obr. 6.3. Každá oblast má následně své vlastní funkce, které vracejí hodnotu hledané termofyzikální veličiny.



Obr. 6.3 Rozdělení modulu IAPWS97 – převzato z [86]

Například pro oblast 1, která charakterizuje kapalně skupenství vody, lze získat tyto fyzikální vlastnosti:

- v : měrný objem [m^3/kg];
- h : měrná entalpie [kJ/kg];
- s : měrná entropie [kJ/kgK];
- c_p : měrná tepelná kapacita při konstantním tlaku [kJ/kgK];
- c_v : měrná tepelná kapacita při konstantním objemu [kJ/kgK];
- w : rychlost zvuku [m/s];
- α_{fav} : tepelná roztažnost [$1/\text{K}$];
- kt : stlačitelnost [$1/\text{MPa}$].

Ukázka použití je v kap. 7.2.

Studijní materiály: Originální dokumentace [87] je velice krátká, dle autorova názoru však bohatě stačí pro pochopení všech funkcí knihovny.

6.5.2. CoolProp

Funkce: Coolprop [88] je knihovna pro výpočet termofyzikálních vlastností velkého množství kapalin a plynů (v roce 2019 122 látek) od podchlazeného stavu až po nadkritické hodnoty. Kromě Pythonu je podporována celá řada programovacích jazyků (C++, JAVA, MATLAB,...) ale i Excel. Knihovna je rozdělena na:

- vlastnosti sytých a nasycených kapalin;
- směsi;
- vlastnosti vlhkého vzduchu;
- nestlačitelné kapaliny;
- vlastnosti vody dle IAPWS97.

Licence: MIT

Použití: Prakticky celé použití knihovny se točí okolo funkce *PropSI*, která se používá pro získání hledané fyzikální vlastnosti za daných parametrů. Např. výpočet entalpie toulenu pro teplotu 293.15 K a suchost páry 0 se dá získat příkazem:

```
PropsSI('Hmolar', 'T', 293.15, 'Q', 0, 'Toluene')
```

Získat lze velké množství fyzikálních vlastností. Jmenovitě například:

- kritickou teplotu;
- dynamickou viskozitu;
- tlak;
- tepelnou vodivost;
- Prandtlovo číslo.

Termodynamické vlastnosti jsou získány derivací Helmholtzovy volné energie [88], jež je definována jako funkce hustoty a teploty. Transportní vlastnosti (viskozita, tepelná vodivost a povrchové napětí) nejsou přítomny pro všechny prvky a vycházejí z experimentálně změřených hodnot, nebo byly použity predikční modely na základě odhadu vlastností podobné látky [88].

Studijní materiály: Dokumentace ke knihovně nemá mnoho stran [88] a měla by stačit pro pochopení knihovny. V případě problému existuje diskuzní skupina na Google Groups [89].

6.5.3. Cantera

Funkce: Cantera [90] je knihovna pro řešení chemické kinetiky a termodynamiky. Knihovna je rozdělena na:

- objekty charakterizující fázi látky;
- termodynamické vlastnosti;
- chemickou kinetiku;
- transportní vlastnosti;
- 0D reaktory;
- 1D reakční toky;
- fyzikální vlastnosti.

Cantera pro jakýkoli výpočet potřebuje importovat data, která definují vlastnosti jednotlivých látek. K tomuto účelu Cantera využívá data ve formátu CTI (Cantera Input), která vznikla převodem standartu CHEMKIN [90].

Cantera podporuje Python, C++ a Matlab.

Licence: BSD

Použití: V této práci byla Cantera použita v příkladu (viz. kap. 7.5) a primárně modul *Solution*, kterým se definuje stav dané směsi a lze jednoduše použít pro jakýkoli typ roztoku, ať už se jedná o směs plynů, kapalin či pevných látek.

Konkrétní použití lze ilustrovat na příkladu níže, kde definujeme plynnou směs (tlak, teplotu, složení).

```
import cantera as ct
```

```
gas = ct.Solution('gri30.cti')
gas.TPX = 298, 101325, 'CH4:1, O2:2'
```

Na Obr. 6.4 je vidět výstup, kde kromě definovaných hodnot knihovna vypočítala i další vlastnosti. S takto zadanou plynnou směsí lze dále pracovat, např. řešit adiabatické spalování pomocí funkce *equilibrate*:

```
gas.equilibrate('TP')
```

Výsledek po chemické reakci lze vidět na Obr. 6.4.

gri30:				gri30:			
temperature	298	K		temperature	298	K	
pressure	101325	Pa		pressure	101325	Pa	
density	1.09107	kg/m ³		density	1.09107	kg/m ³	
mean mol. weight	26.6801	amu		mean mol. weight	26.6801	amu	
	1 kg	1 kmol			1 kg	1 kmol	
enthalpy	-9.322e+05	-2.487e+07	J	enthalpy	-1.0959e+07	-2.924e+08	J
internal energy	-1.0251e+06	-2.735e+07	J	internal energy	-1.1052e+07	-2.949e+08	J
entropy	7652.3	2.042e+05	J/K	entropy	7587	2.024e+05	J/K
Gibbs function	-3.2126e+06	-8.571e+07	J	Gibbs function	-1.322e+07	-3.527e+08	J
heat capacity c_p	1179.9	3.148e+04	J/K	heat capacity c_p	1303.1	3.477e+04	J/K
heat capacity c_v	868.27	2.317e+04	J/K	heat capacity c_v	991.48	2.645e+04	J/K
	X	Y	Chem. Pot. / RT		X	Y	Chem. Pot. / RT
O2	0.666667	0.799567	-25.0791	H2O	0.666667	0.450155	-120.716
CH4	0.333333	0.200433	-53.6221	CO2	0.333333	0.549845	-185.63
[+51 minor]	0	0		[+51 minor]	5.52355e-17	6.62466e-17	

Obr. 6.4 Cantera výstup

V uvedeném příkladu figuruje soubor gri30.cti, který obsahuje základní výběr prvků, a je součástí Cantery. Pro větší výběr látek lze buď vyrobit vlastní soubor CTI (případně upravit již existující), anebo jej převzít např. zde [91].

Studijní zdroje: Originální dokumentace [90] je vhodně napsaná s několika dobrými příklady. Zároveň je to prakticky jediný informační zdroj ke knihovně. Jako doplnění existuje tento tutoriál [92]. V případě problémů se lze obrátit na velmi aktivní Google skupinu [93].

6.6. Knihovna pro práci se soubory

V základu Python umí pracovat pouze s textovými soubory *.txt*. Pro větší výběr je nutné doinstalovat externí knihovny. Jednou z mnoha knihoven může být například knihovna Openpyxl [94].

Funkce: Knihovna je určena pro čtení a zápis excelových souborů *xlsx/xlsm/xltx/xltn* [94]. Obsahuje pouze základní funkce pro úpravu souborů, jako jsou:

- změna hodnot;
- vložení prvku (grafu, obrázku);
- úprava stylu buněk;
- mazání a slučování buněk.

Licence: MIT

Použití: Knihovna umí jen několik základních funkcí, ale její použití je opravdu triviální. Své využití najde v případě, kdy je do již připraveného formuláře nutné doplnit hodnoty. Takový kód by mohl vypadat následovně:

```
import openpyxl

soubor = openpyxl.load_workbook('test.xlsx')
sheet=soubor.active
sheet['A1'] = 2000
soubor.save('test.xlsx')
```

Tento kód do již existujícího dokumentu pojmenovaného jako test.xlsx uloží do buňky A1 hodnotu 2000 a následně uloží do souboru. Dalším příkladem je použití v kap. 7.2.

Studijní zdroje: Pro pochopení knihovny by měla bohatě stačit originální dokumentace [94], která má jen několik málo stran a je přehledně napsaná.

7. Příklady

Tato kapitola volně navazuje na předchozí kapitolu o užitečných Python knihovnách a ilustruje jejich vzájemné použití. Protože tato práce je psána pod záštitou ústavu procesního inženýrství, jsou vybrány příklady z oblasti termomechaniky a hydromechaniky. Pro jistotu, že vše správně funguje, je nutné mít nainstalovány přesné verze knihoven viz příloha č.1.

7.1. Vypouštění vody z nádrže

7.1.1. Zadání:

Určete čas, za který vyteče všechna voda z nádrže. Rozměry nádrže a tvar podstavy udělejte proměnlivé. Uvažujte, že nádrž je shora otevřená a výtokový otvor je výrazně menší v porovnání s rozměry nádrže.

7.1.2. Teoretický rozbor:

Při výpočtu vycházíme z Bernoulliho rovnice [95], která v obecném tvaru je:

$$\frac{v_{kap}^2}{2} + \frac{p}{\rho} + g \cdot z = konst \quad (7.1.1)$$

kde:

v_{kap} - rychlost kapaliny [$m \cdot s^{-1}$]

p - tlak [Pa]

ρ - hustota [$kg \cdot m^{-3}$]

g - gravitační zrychlení [$m \cdot s^{-2}$]

z - tlaková výška [m]

Tu následně spočítáme pro body 1 a 2 viz Obr. 7.1, které dáme do vzájemné rovnosti:

$$\frac{v_1^2}{2} + \frac{p_1}{\rho} + g \cdot z_1 = \frac{v_2^2}{2} + \frac{p_2}{\rho} + g \cdot z_2 \quad (7.1.2)$$

Tento vztah platí pro ideální kapalinu. V případě skutečné kapaliny má rovnice tvar[95]:

$$\frac{v_1^2}{2} + \frac{p_1}{\rho} + g \cdot z_1 = \frac{v_2^2}{2} + \frac{p_2}{\rho} + g \cdot z_2 + Y_{z\ 1,2} \quad (7.1.3)$$

kde:

$Y_{z\ 1,2}$ - měrná ztrátová energie [$J \cdot kg^{-1}$]

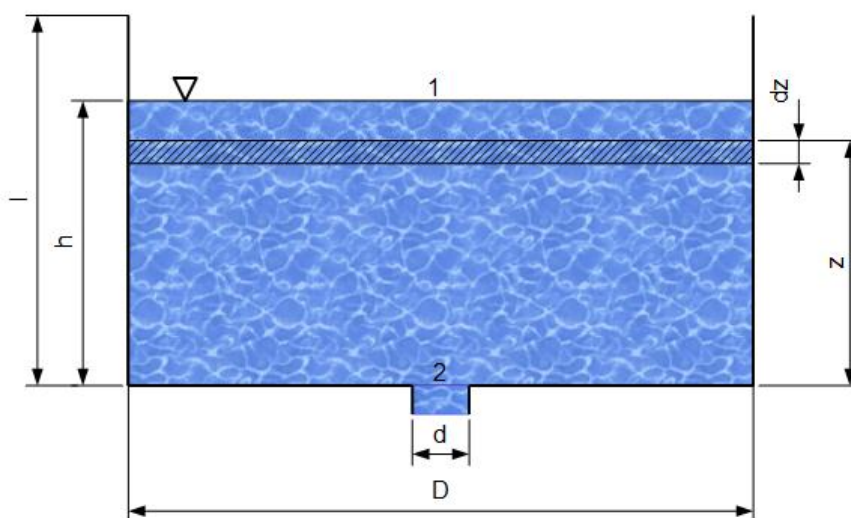
Ta dle Weisbachova vztahu [95] se vypočte jako:

$$Y_{z\,1,2} = \xi \frac{v_2^2}{2} \quad (7.1.4)$$

Kde:

ξ - ztrátový součinitel výtokového otvoru [-]

Ztrátový součinitel otvoru nelze vypočítat a je nutné jej určit experimentálně [95].



Obr. 7.1 Schéma nádrže

Dále uvažujeme že:

- V momentě otevření vypouštěcího ventilu je rychlost na hladině nulová:

$$v_1 = 0 \quad (7.1.5)$$

- Nádrž je otevřená, tudíž je stejný tlak jak na hladině, tak i na výstupu v bodě 1:

$$p_1 = p_2 \quad (7.1.6)$$

- Výška v bodě jedna je nulová:

$$z_1 = 0 \quad (7.1.7)$$

- Přítok kapaliny do nádrže je nulový.
- Kapalinu považujeme za neviskózní.

Díky těmto předpokladům vyjde z Bernouliho rovnice vztah pro teoretickou výtakovou rychlost:

$$0 + 0 + g \cdot z_1 = \frac{v_2^2}{2} + 0 + 0 + \xi \frac{v_2^2}{2} \quad (7.1.8)$$

$$v_2 = \frac{1}{\sqrt{1 + \xi}} \sqrt{2gz_1} \rightarrow \varphi \sqrt{2gz_1} \quad (7.1.9)$$

Kde:

φ - rychlostní součinitel [-]

Tuto rychlost použijeme pro určení výtoku pomocí elementárního objemu [95]:

$$Q = \frac{dV}{dt} \rightarrow dV = Q \cdot dt = S \cdot v_{kap} \cdot dt \quad (7.1.10)$$

Kde:

Q - průtok [$\text{m}^3 \cdot \text{s}^{-1}$]

Ten nahradíme jako [95]:

$$\underline{dV = -S \cdot dz} \quad (7.1.11)$$

$$-S_1 \cdot dz = \varphi \cdot S_2 \cdot \sqrt{2 \cdot g \cdot z} \cdot dt \quad (7.1.12)$$

Kde:

S_1 - průřez nádrže [m^2]

S_2 - průřez výtoku [m^2]

Následně získáme závislost času na změně hladiny [95]:

$$dt = - \frac{S_1}{\varphi \cdot S_2 \cdot \sqrt{2 \cdot g}} \cdot \frac{dz}{\sqrt{z}} \quad (7.1.13)$$

Výsledný čas dostaneme integrací obou stran:

$$T = \int_0^T dt = - \frac{S_1}{\varphi \cdot S_2 \cdot \sqrt{2 \cdot g}} \cdot \int_h^0 \frac{dz}{\sqrt{z}} \quad (7.1.14)$$

$$T = \frac{2 \cdot S_1}{\varphi \cdot S_2 \cdot \sqrt{2 \cdot g}} \cdot \sqrt{h} \quad (7.1.15)$$

Kde:

T - čas [s]

h - výška hladiny [m]

7.1.3. Programové řešení:

Použité knihovny:

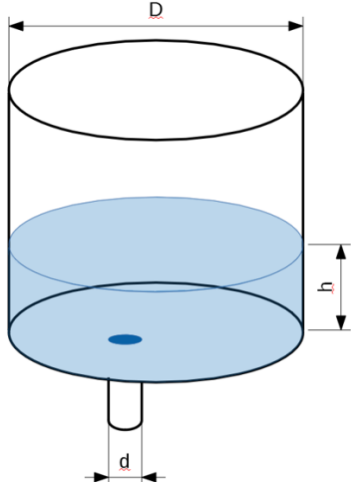
- Jupyter Widgets
- Scipy

Použité nástroje:

- Jupyter Notebook

Program je rozdělen ve dvou buňkách Jupyter Notebooku. První buňka obsahuje importované balíčky a kód, který definuje uživatelské prostředí vytvořené pomocí Jupyter Widgets. Druhá buňka vzájemně propojuje widgety a obsahuje všechny funkce nutné pro výpočet. Výsledná spolupráce těchto buněk slouží k vykreslení GUI (viz Obr. 7.2).

Tvar podstavy:	<input checked="" type="radio"/> Kruh <input type="radio"/> Obdélník
Průměr D [m]:	<input type="text" value="0"/>
Průměr výtokového hrdla d [m]:	<input type="range" value="0.00"/> <input type="text" value="0"/>
Výška hladiny h [m]:	<input type="text" value="0"/>
Ztrátový součinitel otvoru: [-]	<input type="text" value="0.64"/>
Čas vypouštění nádrže t [s]:	<input type="text" value="0"/>



Obr. 7.2 Uživatelské prostředí pomocí Jupyter Widget

GUI umožňuje zvolit tvar podstavy nádrže a parametry nutné pro výpočet času. Hodnoty průměru (nebo velikosti stran) a výšky hladiny mají omezení v intervalu 0 až 1000. Průměr výtokového hrdla má omezení v podobě 1/10 průměru a v případě obdélníkové podstavy je omezení 1/10 z menší strany. Je to z důvodu, aby byl dodržen předpoklad, že nádrž je výrazně větší než výtokový otvor. Ztrátový součinitel je defaultně nastavený na hodnotu 0,64, ale lze jej v zadání přepsat.

7.1.4. Kontrola výpočtu:

Program není napsaný pro řešení konkrétního zadání. Proto kontrola programu je ve formě vlastního zadání:

Válcová nádrž o průměru 1,5 metrů je do výšky 5 metrů naplněna vodou. Průměr výpusti je 10 cm. Ztrátový součinitel na výtoku je 0,64. Za jak dlouho se nádrž vypustí?

Výpočet obsahů:

$$S_1 = \frac{\pi \cdot d_1^2}{4} = \frac{\pi \cdot 1.5^2}{4} = 1.767 \text{ m}^2$$
$$S_2 = \frac{\pi \cdot d_2^2}{4} = \frac{\pi \cdot 0.1^2}{4} = 7.85 \cdot 10^{-3} \text{ m}^2$$

Výpočet rychlostního součinitele:

$$\varphi = \frac{1}{\sqrt{1 + \xi}} = \frac{1}{\sqrt{1 + 0.64}} = 0.781$$

Výpočet času:

$$T = \frac{2S_1}{\varphi \cdot S_2 \sqrt{2g}} \sqrt{h} = \frac{2 \cdot 1.767}{0.781 \cdot 7.85 \cdot 10^{-3} \sqrt{2 \cdot 9.81}} \sqrt{5} = 290.917 \text{ s}$$

Program vypočítal pro stejné hodnoty identický čas.

7.2. Vypařování vody

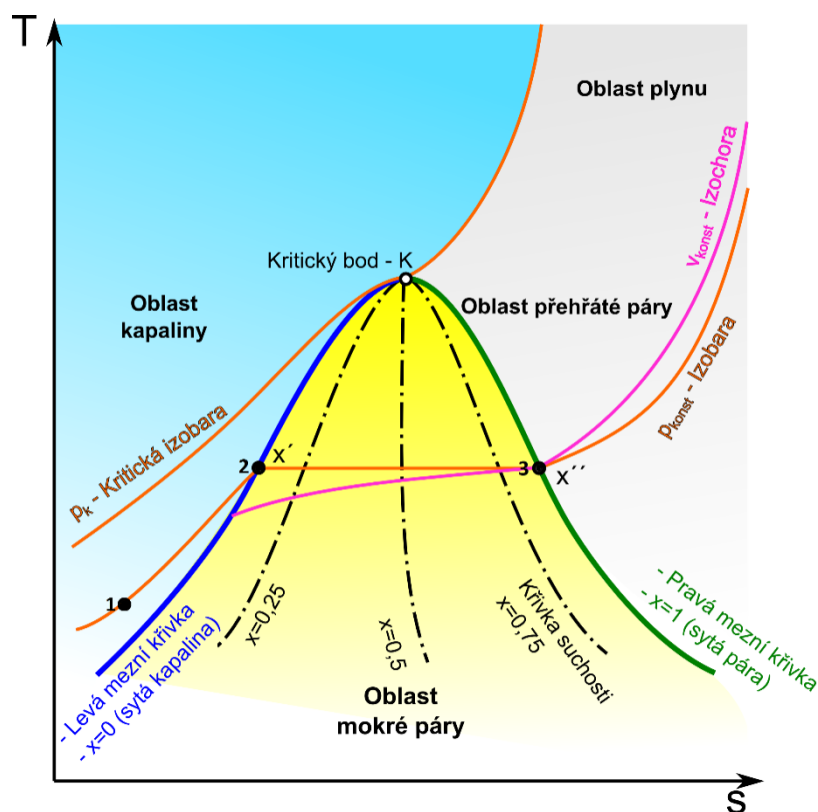
7.2.1. Zadání:

Určete množství tepla, které se musí dodat vodě, aby vznikla přehřátá pára. Pohybujte se pod kritickým bodem vody.

7.2.2. Teorie:

Celý proces výroby páry z vody se skládá ze dvou výpočtů. První výpočet je pro stanovení tepla potřebného na přeměnu vody na tzv. sytou kapalinu. To je hranice, kde se stýká oblast kapaliny s oblastí mokré páry. Mokrý pára je směs vody a páry a jejich poměr je definován tzv. suchostí – v literatuře označené jako „x“. Druhým krokem je výpočet tzv. výparného tepla (neboli latentního), což je teplo potřebné na změnu ze syté kapaliny na sytou páru, jde o stav, kdy se všechna voda vypaří. Oba tyto děje se dějí za konstantního tlaku, tudíž uvažujeme izobarický děj [96].

Názorně jsou tyto oblasti vidět na T-s diagramu vodní páry (viz Obr. 7.3).



Obr. 7.3 T-s diagram vody – převzato z [97]

Teplo potřebné pro výrobu syté kapaliny se určí pomocí druhé formy 1. termodynamického zákona [96]:

$$q = dh - v dp \quad (7.2.1)$$

Díky izobarickému ději nám vypadne člen s rozdílem tlaků a vznikne nám:

$$q_{12} = h_2 - h_1 \quad (7.2.2)$$

Kde:

q_{12} - měrné teplo [$\text{kJ} \cdot \text{kg}^{-1}$]

h_1 - měrná entalpie kapaliny [$\text{kJ} \cdot \text{kg}^{-1}$]

h_2 - měrná entalpie syté kapaliny [$\text{kJ} \cdot \text{kg}^{-1}$]

Díky tomu, že přeměna ze syté kapaliny na sytou páru probíhá také izobaricky [96], pro výparné teplo použijeme stejný vzorec:

$$q_{23} = h_3 - h_2 \quad (7.2.3)$$

Kde:

q_{23} - měrné teplo [$\text{kJ} \cdot \text{kg}^{-1}$]

h_3 - měrná entalpie syté páry [$\text{kJ} \cdot \text{kg}^{-1}$]

Hodnoty entalpie v oblasti syté kapaliny lze vyjádřit jako [96]:

$$h_x = h_2 + x \cdot (h_3 - h_2) \quad (7.2.4)$$

Kde:

x - suchost páry [-]

h_x - měrná entalpie mokré páry [$\text{kJ} \cdot \text{kg}^{-1}$]

Podobně jako entalpie je stejný výpočet i pro entropii a měrný objem [96]:

$$s_x = s_2 + x \cdot (s_3 - s_2) \quad (7.2.5)$$

$$v_x = v_2 + x \cdot (v_3 - v_2) \quad (7.2.6)$$

Kde:

s_x - měrná entropie mokré páry [$\text{kJ} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$]

s_2 - měrná entropie syté kapaliny [$\text{kJ} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$]

s_3 - měrná entropie syté páry [$\text{kJ} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$]

v_x - měrný objem mokré páry [$\text{m}^3 \cdot \text{kg}^{-1}$]

v_2 - měrný objem syté kapaliny [$\text{m}^3 \cdot \text{kg}^{-1}$]

v_3 - měrný objem syté páry [$\text{m}^3 \cdot \text{kg}^{-1}$]

Suchost páry vychází ze vztahu [96]:

$$x = \frac{m''}{m' + m''} \quad (7.2.7)$$

Kde:

m'' - hmotnost syté páry [kg]

m' - hmotnost syté kapaliny [kg]

Celkové měrné teplo pro přeměnu vody na sytou páru získáme:

$$q = q_{12} + q_{23} \quad (7.2.8)$$

Kde:

q - měrné celkové teplo [$\text{kJ} \cdot \text{kg}^{-1}$]

Jaký výkon je nutné dodat výparníku, je dáno výpočtem:

$$P = q \cdot \dot{m} \quad (7.2.9)$$

Kde:

P - výkon [kW]

\dot{m} - hmotnostní tok [$\text{kg} \cdot \text{s}^{-1}$]

7.2.3. Programové řešení:

Použité knihovny:

- IAPWS
- PyQt
- Openpyxl

Použité nástroje:

- Qt Designer
- Excel

Program pro získání hodnot vody a vodní páry využívá knihovnu IAPWS a konkrétně regiony 1 a 4 dle standartu IAPWS-97. Region 1 je použit pro získání vlastností vody a region 4 pro hledání hodnot syté páry a syté kapaliny (body 2 a 3 viz Obr. 7.3) a pro kontrolu, jestli vstupní hodnoty nejsou v oblasti přehřáté páry. Celý výpočet je zabalený do GUI vytvořeném v PyQt.

Program je rozdělený do dvou Python souborů. První soubor pod názvem *gui.py* obsahuje kód, který definuje vzhled uživatelského prostředí. Druhý soubor obsahuje kód, který definuje všechny funkce nutné pro chod programu a zároveň se jím program spouští.

Návod k programu viz příloha č. 2.

7.2.4. Kontrola výpočtu:

Kontrolu výpočtu provedeme pomocí tabulek IAPWS-97 uvedených v [96] pro hodnoty tlaku 12 MPa a počáteční teplotu vody 200 °C. Z Tab. 7.1 je vidět, že program vychází z identických hodnot.

Tab. 7.1 Porovnání vstupních dat pro výpočet výparníku

		Hodnoty z programu	Hodnoty z tabulek
Měrné entalpie [kJ/kg]	h1	856,8	856,8
	h2	1491,3	1491,3
	h3	2685,6	2685,6
Měrný objem [m ³ /kg]	v1	0,00115	0,00115
	v2	0,00153	0,00153
	v3	0,01427	0,01427
Měrné entropie [kJ/kg*K]	s1	2,3147	2,3147
	s2	3,4965	3,4965
	s3	5,4941	5,4941
Teplota [°C]			
	t3	324,68	324,68

$$q = h_2 - h_1 + h_3 - h_2 = 1491.3 - 856.8 + 2685.6 - 1491.3 = 1828.8 \text{ kJ/kg}$$

$$P = \dot{m} \cdot q = 10 \cdot 1828.8 = 18288 \text{ kW}$$

Vypočtené hodnoty jsou identické (viz příloha č. 2.)

7.3. Tepelné ztráty zateplené stěny

7.3.1. Zadání:

Určete tepelné ztráty zateplené stěny a porovnejte časovou návratnost investice. Zateplená stěna je z železobetonu a na vnitřní straně je omítnuta vápennou omítkou, izolace z vnější strany je omítnuta perlitovou omítkou.

Průměrná teplota uvnitř	$T_{in} = 22 \text{ °C}$
Průměrná teplota venku	$T_{out} = 6 \text{ °C}$
Plocha stěny	$S = 200 \text{ m}^2$
Tepelná vodivost železobetonu	$\lambda_{BET} = 1,43 \text{ W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$
Tepelná vodivost perlitové omítky	$\lambda_{PER} = 0,1 \text{ W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$
Tepelná vodivost vápenné omítky	$\lambda_{VAP} = 0,88 \text{ W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$
Součinitel přestupu tepla na vnitřní straně [100]	$\alpha_{in} = 7 \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$
Součinitel přestupu tepla na vnější straně [100]	$\alpha_{out} = 17 \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$
Tloušťka stěny	$l_{BET} = 0,2 \text{ m}$

Tloušťka izolace	$l_{IZOL} = 0,1 \text{ m}$
Tloušťka perlitové omítky	$l_{PER} = 0,015 \text{ m}$
Tloušťka vápenné omítky	$l_{VAP} = 0,015 \text{ m}$
Cena plynu	$C_{plyn} = 4 \text{ Kč} \cdot \text{kWh}^{-1}$

Tepelná vodivost a cena 1m^2 izolace viz [98].

7.3.2. Teorie:

Při výpočtu tepelných ztrát se používá analogie na Ohmův zákon. Tedy platí [99]:

$$I = \frac{U}{R} \rightarrow \dot{Q} = \frac{\Delta T}{R} \quad 7.3.1$$

Kde:

\dot{Q} - tepelný tok [W]

ΔT - rozdíl teplot [K]

R - tepelný odpor [$\text{K} \cdot \text{W}^{-1}$]

Tepelný odpor lze dále vyjádřit jako [99]:

$$R = \frac{\delta}{\lambda \cdot S} \quad 7.3.2$$

Kde:

δ - šířka stěny [m]

λ - tepelná vodivost [$\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$]

S - plocha stěny [m^2]

Zvážíme-li vliv přestupu tepla i mezi vzduchem a stěnou, výpočet odporu je [99]:

$$R = \frac{1}{\alpha \cdot S} \quad 7.3.3$$

Kde:

α - součinitel přestupu tepla [$\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$]

Pokud se stěna skládá z více vrstev, tak se jednotlivé odpory sčítají, podobně jako tomu je u sériově zapojeného odporu. Celkový tepelný tok stěnou je pak dán vztahem [99]:

$$\dot{Q} = \frac{\Delta T}{\frac{1}{\alpha_{in} \cdot S} + \sum_{i=1}^n \frac{\delta}{\lambda \cdot S} + \frac{1}{\alpha_{out} \cdot S}} \quad 7.3.4$$

Kde:

α_{in} - součinitel přestupu tepla na vnitřní straně stěny [$\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$]

α_{out} - součinitel přestupu tepla na vnější straně stěny [$\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$]

ΔT - rozdíl teplot [K]

\dot{Q} - tepelné ztráty [W]

Při výpočtu návratnosti investice se vychází z rovnice, kde na jedné straně je cena izolace (zanedbáváme stavební náklady) spolu s náklady na vytápění zateplené stěny a na druhé straně náklady na vytápění nezateplené stěny [100]:

$$C_{izol} + \dot{Q}_{izol} \cdot C_{plyn} \cdot t_{inv} = \dot{Q}_{neizol} \cdot C_{plyn} \cdot t_{inv} \quad 7.3.5$$

$$t_{inv} = \frac{C_{izol}}{C_{plyn} \cdot (\dot{Q}_{neizol} - \dot{Q}_{izol})} \quad 7.3.6$$

Kde:

C_{izol} - cena materiálu izolace [Kč]

C_{plyn} - cena plynu [$\text{Kč} \cdot \text{kWh}^{-1}$]

\dot{Q}_{izol} - tepelné ztráty izolovanou stěnou [kW]

\dot{Q}_{neizol} - tepelné ztráty neizolovanou stěnou [kW]

t_{inv} - návratnost investice [hod]

7.3.3. Programové řešení:

Použité knihovny:

- Pandas;
- Numpy;
- xlrd (doplněk Pandas pro práci s *.xlsx* soubory)
- lxml (doplněk Pandas pro práci s *.html* soubory)

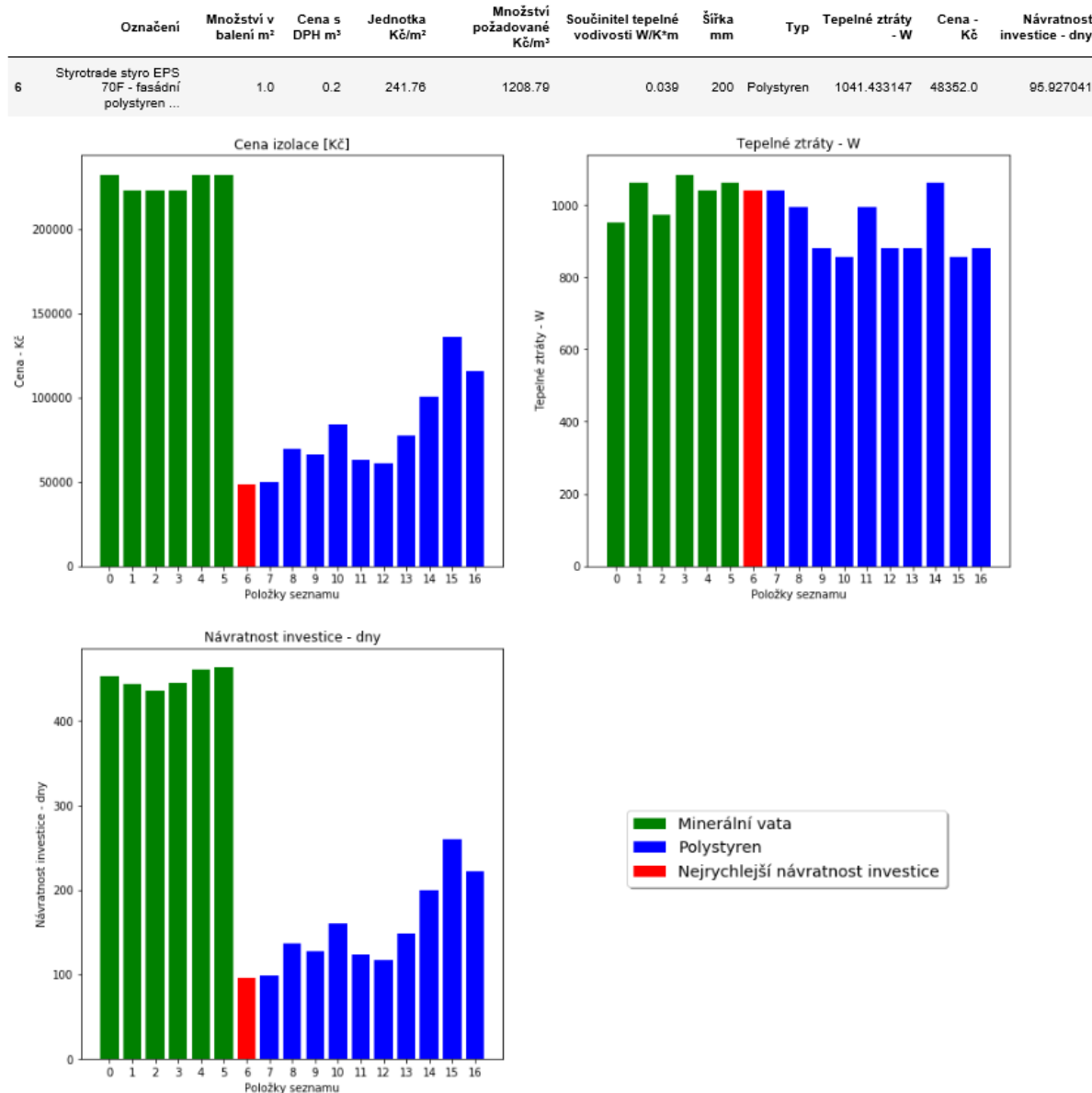
Použité nástroje:

- Excel;
- Jupyter Notebook.

Výpočet tepelných ztrát je rozdělen do 4 buněk Jupyter Notebooku který se stará o výpočet a souboru *data.py*, který se stará o zpracování dat z [98] a ukládá je do souboru *data.xlsx*. Buňky se starají o:

1. V první buňce jsou importované všechny knihovny a definované parametry výpočtu. Pro představu, co které proměnná znamená, je pod touto buňkou načtená ilustrace. Hodnota tepelné vodivosti betonu a omítek je z [101].

2. Následující buňka slouží pro načtení dat ze souboru *data.xlsx*. Ty byly získány pomocí programu *data.py*, který načítá data ze staticky uložených webových stránek z portálu [98] a ukládá je do excelu. Zároveň opravuje některé chybové údaje.
3. Třetí buňka filtruje data načtená podle požadované tloušťky izolace a přidává sloupce *Tepelné ztráty – W*, *Cena – Kč* a sloupec *Návratnost investice – dny*.
4. Poslední buňka vypíše izolaci s co nejmenší hodnotou návratnosti a zobrazí ji v kontextu zbylých izolací pomocí trojce grafů (viz Obr. 7.4).



Obr. 7.4 Srovnání vybraných izolací

7.3.4. Kontrola výpočtu:

Kontrola výpočtu viz hodnoty ze zadání a hodnota tepelné vodivosti spolu s cenou za m² z polystyrenu, který se umístil na prvním místě, z pohledu návratnosti investice viz Obr. 7.4

Tepelný odpor zateplené stěny:

$$R_{IZOL} = \frac{1}{\alpha_{IN} \cdot S} + \frac{l_{PER}}{\lambda_{PER} \cdot S} + \frac{l_{BET}}{\lambda_{BET} \cdot S} + \frac{l_{IZOL}}{\lambda_{IZOL} \cdot S} + \frac{l_{VAP}}{\lambda_{VAP} \cdot S} + \frac{1}{\alpha_{OUT} \cdot S} =$$

$$= \frac{1}{7 \cdot 200} + \frac{0,015}{0,1 \cdot 200} + \frac{0,2}{1,43 \cdot 200} + \frac{0,1}{0,039 \cdot 200} + \frac{0,015}{0,88 \cdot 200} + \frac{1}{17 \cdot 200} = 0,00254 \text{ K/W}$$

Tepelný odpor nezateplené stěny:

$$R_{NEIZOL} = \frac{1}{\alpha_{IN} \cdot S} + \frac{l_{PER}}{\lambda_{PER} \cdot S} + \frac{l_{BET}}{\lambda_{BET} \cdot S} + \frac{l_{VAP}}{\lambda_{VAP} \cdot S} + \frac{1}{\alpha_{OUT} \cdot S} =$$

$$= \frac{1}{7 \cdot 200} + \frac{0,015}{0,1 \cdot 200} + \frac{0,2}{1,43 \cdot 200} + \frac{0,015}{0,88 \cdot 200} + \frac{1}{17 \cdot 200} = 0,00153 \text{ K/W}$$

Tepelné ztráty zateplené stěny:

$$\dot{Q}_{izol} = \frac{\Delta T}{R_{IZOL}} = \frac{295,15 - 279,15}{0,0164} = 1,041 \text{ kW}$$

Tepelné ztráty nezateplené stěny:

$$\dot{Q}_{neizol} = \frac{\Delta T}{R_{NEIZOL}} = \frac{295,15 - 279,15}{0,0164} = 6,292 \text{ kW}$$

Cena izolace:

$$C_{izol} = C_m \cdot S = 241,76 \cdot 200 = 48352 \text{ Kč}$$

Čas návratnosti investice:

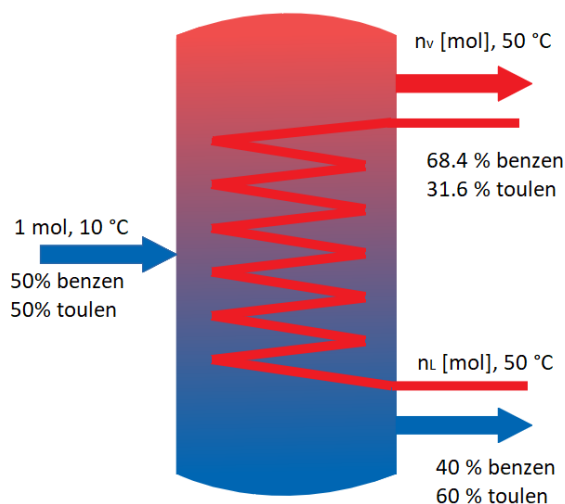
$$t_{inv} = \frac{C_{izol}}{C_{plyn} \cdot (\dot{Q}_{neizol} - \dot{Q}_{izol})} = \frac{48352}{4 \cdot (6,292 - 1,041)} = 2302,037 \text{ hod} = 95,92 \text{ dnu}$$

Program vypočítal identickou hodnotu (viz Obr. 7.4).

7.4. Částečné vypaření směsi

7.4.1. Zadání

Směs benzenu a toluenu je přiváděna do výparníku, ve kterém se ohřívá na 50 °C. Tato směs je na vstupu tvořena 50% benzenem a 50% toluenem. Na výstupu výpary obsahují 68,4 % benzenu a kapalný zbytek obsahuje 40 % benzenu. Zbytek tvoří toluen. Jaké teplo je nutné průběžně dodávat do výparníku? Zanedbejte tlak na vstupu do výparníku [102].



Obr. 7.5 Výpařování směsi

7.4.2. Teorie

Molární hmotnosti benzenu a toluenu zjistíme z řešení rovnic:

$$1 \text{ mol} = n_p + n_k \quad 7.4.1$$

$$0.5 \text{ mol} = 0.684 n_{p \text{ benz}} + 0.4 n_{k \text{ benz}}$$

$$0.5 \text{ mol} = 0.316 n_{p \text{ toul}} + 0.6 n_{k \text{ toul}}$$

Řešením lineárních rovnic pomocí matice dostaneme:

$$\left(\begin{array}{cc|c} 1 & 1 & 1 \\ 0.684 & 0.4 & 0.5 \end{array} \right) = \left(\begin{array}{cc|c} 1 & 1 & 1 \\ 0 & 0.284 & 0.184 \end{array} \right)$$

$$0.284 n_k = 0.184 \rightarrow n_k = 0.6479$$

$$n_p + 0.6479 = 1 \rightarrow n_p = 0.3521$$

$$n_{p \text{ benz}} = 0.684 n_p = 0.2408 \text{ mol}$$

$$n_{k \text{ benz}} = 0.40 n_k = 0.2592 \text{ mol}$$

$$n_{p \text{ toul}} = 0.316 n_p = 0.1113 \text{ mol}$$

$$n_{k \text{ toul}} = 0.60 n_k = 0.3887 \text{ mol}$$

Kde:

n_p - molární množství páry [mol]

n_k - molární množství kapaliny [mol]

Protože výparník funguje při stejném tlaku, výpočet tepla je rozdíl entalpií výstupních a vstupních položek:

$$Q = \Delta H = \sum_{out} h_i n_i - \sum_{in} h_i n_i \quad 7.4.2$$

Kde:

h_i - měrná entalpie složky [$\text{J} \cdot \text{mol}^{-1}$]

Q - teplo [J]

Jednotlivé entalpie jsou vypočteny z derivace Helmholtzovy volné energie získané z knihovny Coolprop.

7.4.3. Program:

Použité knihovny:

- CoolProp;
- Numpy.

Použité nástroje:

- Jupyter.

Kód je rozdělen do dvou Jupyter buněk. V první je řešen výpočet jednotlivých složek směsi na základě matice řešené knihovnou Numpy pomocí funkce *linalg*. Druhá počítá výsledné teplo nutné pro chod výparníku z měrných entalpií získaných knihovnou CoolProp pomocí funkce *PropSI*.

7.4.4. Kontrola výpočtu:

Zadání příkladu je převzato z [102], proto jen porovnáme výsledky:

Tab. 7.2 Porovnání hodnot – částečné vypařování

	kniha	program	% odchylka od knihy
dh1 výparné složky toulenu [J/mol]	6340	6332	-0,126
dh2 kapalně složky toulenu [J/mol]	42930	43012	0,191
dh3 výparné složky benzenu [J/mol]	5332	5470	2,588
dh4 kapalně složky benzenu [J/mol]	37520	37932	1,098
Celkové teplo [J]	17700	17800	0,565

Odchylka kolem 2,5 % u výparné složky benzenu je způsobena rozdílným zdrojem dat. U knihy [102] není přímo definováno, odkud jsou čerpány data pro výpočet entalpie. Na druhou stranu CoolProp uvádí konkrétní odkaz na zdroj dat pro toulenu [103] a benzen [104].

7.5. Spalování plynné směsi

7.5.1. Zadání

Do hořáku proudí 1500 m³/hod. zemního plynu. Jako okysličovadlo je použit atmosférický vzduch o relativní vlhkosti 50%. Teplota okysličovadla i plynu je 40 °C a míchání probíhá za atmosférického tlaku (101325 Pa).

Objemové složení zemního plynu je:

- Methan (CH₄) = 85,8 %
- Ethan (C₂H₆) = 8,49 %
- Propan (C₃H₈) = 2,3 %
- Butan (C₄H₁₀) = 0,7 %
- Pentan (C₅H₁₂) = 0,25 %
- Dusík (N₂) = 0,96 %
- Oxid uhličitý (CO₂) = 1,5 %

Objemové složení suchého vzduchu:

- Dusík (N₂) = 78,04 %
- Kyslík (O₂) = 20,99 %
- Argon (Ar) = 0,94 %
- Oxid uhličitý (CO₂) = 0,03 %

7.5.2. Teoretický rozbor:

Na začátku je potřeba definovat:

- Plyny rozdělujeme na [105]:

Ideální – U ideálních plynů platí, že molekuly plynů jsou vzájemně hodně vzdálené, lze tedy zanedbat vliv ostatních molekul (přitažlivé a odpuzivé síly) a lze zanedbat jejich velikost. Plyny lze za ideální považovat za nízkého tlaku nebo vysoké teploty.

Reálné – Reálné plyny jsou opakem ideálních a je nutné počítat i se vzájemným ovlivňováním částic a velikostí molekul. Plyny jako reálné je nutné považovat okolo hodnot blízkých zkapalnění nebo blízko kritického bodu.

Pro potřeby výpočtu spalování, kdy se pohybujeme okolo normálních teplot a za tlaků blízkých atmosférickému, považujeme plyny jako ideální. Ideální plyny se řídí stavovou rovnicí [96]:

$$\frac{p \cdot v}{T} = r \quad 7.5.1$$

Kde:

p - tlak [Pa]

v - měrný objem [m³·kg⁻¹]

T - teplota [K]

r - měrná plynová konstanta [$\text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$]

A dále platí [96]:

$$r = \frac{R_m}{M}, v = \frac{V}{m}, M = \frac{m}{n}, V = n \cdot N_V \quad 7.5.2$$

Kde:

R_m - univerzální plynová konstanta [$8,3143 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$]

V - objem [m^3]

m - hmotnost [kg]

M - molární hmotnost [$\text{kg} \cdot \text{kmol}^{-1}$]

n - molární množství [kmol]

N_V - molární objem = $22,414 \text{ [m}^3 \cdot \text{kmol}^{-1}]$

Pro stanovení jednotlivých složek ve směsi lze využít některého vztahu [96]:

$$w_i = \frac{m_i}{m} \rightarrow c_{wi} = w_i \cdot 100 \quad 7.5.3$$

$$x_i = \frac{n_i}{n} = \frac{V_i}{V} = \frac{p_i}{p} \rightarrow c_{ni} = x_i \cdot 100 \quad 7.5.4$$

Kde:

w_i - hmotnostní zlomek [-]

m_i - hmotnost dané složky [kg]

c_{wi} - hmotnostní koncentrace [%]

x_i - molární/objemový zlomek [-]

n_i - molární množství dané složky [kmol]

V_i - objem dané složky [m^3]

p_i - parciální tlak dané složky [Pa]

c_{ni} - molární/objemová koncentrace [%]

- Předpokládáme dokonalé spalování. To znamená, že množství kyslíku pro spálení paliva je dáno pomocí stechiometrických koeficientů [105].

Výpočet následně rozdělíme do několika částí:

- **Složení vlhkého vzduchu**

Vzduch rozlišujeme na tzv. suchý a vlhký. Vlhký vzduch na rozdíl od suchého obsahuje malé množství vodní páry, která se změnou teploty a tlaku kolísá.

Díky předpokladu, že se pohybujeme okolo atmosférického tlaku a běžných teplot, můžeme vlhký vzduch počítat jako ideální směs plynů. Proto koncentraci vody ve vzduchu spočítáme jako [106]:

$$\varphi = \frac{p_{H_2O}}{p_{H_2O}''} = \frac{\frac{p_{H_2O}}{p}}{\frac{p_{H_2O}''}{p}} = \frac{c_{VH_2O}}{c_{VH_2O}''} \rightarrow c_{VH_2O} = \varphi \cdot c_{VH_2O}''$$

$$= \varphi \cdot \frac{p_{H_2O}''}{p}$$
7.5.5

Kde:

φ - relativní vlhkost [-]

p_{H_2O} - parciální tlak vodní páry [Pa]

p_{H_2O}'' - parciální tlak nasycené vodní páry [Pa]

c_{VH_2O} - objemová koncentrace vodní páry [%]

c_{VH_2O}'' - objemová koncentrace nasycené vodní páry [%]

Tlak nasycené vodní páry lze zjistit buď pomocí tabulek, či pomocí Antoinetovy rovnice [96].

Složení vlhkého vzduchu pak spočteme jako:

$$c_i^{VV} = c_i^{SV} \cdot (1 - c_{VH_2O})$$
7.5.6

• Množství kyslíku

Chemické reakce při spalování jsou popsány pomocí stechiometrických rovnic, z kterých lze určit množství prvků vystupujících v chemické reakci [105]. Například z rovnice:



Z rovnice vyplývá, že pro spálení 1 molu vodíku je potřeba 0.5 molu kyslíku. Při výpočtu směsi je výpočet následující [106]:

$$\dot{n}_{O_2} = \sum_{i=1}^n c_{ni} \cdot S_{iO_2} \cdot \dot{n}^P$$
7.5.8

Kde:

\dot{n}_{O_2} - molový průtok kyslíku [kmol·hod⁻¹]

S_{iO_2} - stechiometrický koeficient udávající množství kyslíku [-]

\dot{n}^P - molový průtok paliva [kmol·hod⁻¹]

Celkové množství oksyličovadla se vypočítá pomocí vztahu:

$$\dot{n}^{OK} = \frac{\dot{n}_{O_2}}{c_{nO_2}^{OK}} \quad 7.5.9$$

Kde:

\dot{n}^{OK} - molový tok oksyličovadla [kmol·hod⁻¹]

$c_{nO_2}^{OK}$ - molární koncentrace kyslíku v oksyličovadle [%]

Poměr mezi množstvím paliva a oksyličovadla udává tzv. air/fuel ratio, které vypočteme dle vztahu [105]:

$$AFR = \frac{\dot{n}^{OK}}{\dot{n}^P} \quad 7.5.10$$

Kde:

AFR - poměr mezi množstvím oksyličovadla a paliva [-]

\dot{n}^P - molový tok paliva [kmol·hod⁻¹]

- **Složení spalin**

Podobným způsobem, jakým jsme vypočítali množství kyslíku potřebného pro ideální spalování, lze určit ze stechiometrických rovnic výsledné složení spalin.

- **Průtoky spalin**

Spaliny lze rozdělit na suché a vlhké. Vlhké spaliny jsou produktem spalování paliva a oksyličovadla [105]. Platí tedy:

$$\dot{m}^{VSP} = \dot{m}^{OK} + \dot{m}^P \quad 7.5.11$$

Kde:

\dot{m}^{VSP} - hmotnostní tok vlhkých spalin [kg·hod⁻¹]

\dot{m}^{OK} - hmotnostní tok oksyličovadla [kg·hod⁻¹]

\dot{m}^P - hmotnostní tok paliva [kg·hod⁻¹]

Suché spaliny vzniknou z vlhkých spalin odstraněním vodní páry [105]:

$$\dot{m}^{SSP} = \dot{m}^{VSP} \cdot (1 - c_{wH_2O}^{VSP}) \quad 7.5.12$$

Kde:

\dot{m}^{SP} - hmotnostní tok suchých spalin [$\text{kg} \cdot \text{hod}^{-1}$]

$c_{wH_2O}^{VSP}$ - hmotnostní koncentrace vody ve vlhkých spalínách [%]

• Entalpie plynů

Jednotlivé entalpie se počítají dle NASA polynomu [107][108]:

$$h_{T-0} = R \cdot T \left(A_1 + A_2 \cdot \frac{T}{2} + A_3 \cdot \frac{T^2}{3} + A_4 \cdot \frac{T^3}{4} + A_5 \cdot \frac{T^4}{5} + A_6 \cdot \frac{1}{T} \right) \quad 7.5.13$$

Kde:

h_{T-0} – měrná entalpie vztažená k 0 K [$\text{kJ} \cdot \text{mol}^{-1}$]

R - univerzální plynová konstanta [$\text{J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$]

$A_1, A_2, A_3, A_4, A_5, A_6$ - experimentálně získané hodnoty pro konkrétní prvek [-]

Takto vypočtená entalpie je vztažená k referenční teplotě 0 K. Pro výpočet entalpie vztažené k jiné referenční teplotě, než je 0 K, je výpočet následující [108]:

$$h_{T-273.15} = h_{273.15-0} + (h_{T-0} - h_{273.15-0}) \quad 7.5.14$$

Kde:

$h_{273.15-0}$ - hodnota měrné entalpie pro 273.15 K vztažená k 0 K [$\text{kJ} \cdot \text{mol}^{-1}$]

• Výhřevnost a spalné teplo

Výhřevnost (lower heating value – LHV) je teplo uvolněné dokonalým spálením 1kg paliva s kyslíkem, jehož spaliny jsou ochlazené na původní teplotu paliva. Voda obsažená ve spalínách zůstane v plynném stavu [105]. Výpočet je dle [105]:

$$l hv = \sum_{i=1}^n n_i^{VSP} \cdot h_i^{VSP} - \sum_{i=1}^n n_i^P \cdot h_i^P \quad 7.5.15$$

$$m^P = \sum_{i=1}^n n_i^P \cdot M_i \quad 7.5.16$$

$$LHV = \frac{l hv}{m^P} \quad 7.5.17$$

Kde:

lhv - výhřevnost [$\text{kJ} \cdot \text{mol}^{-1}$]

LHV - výhřevnost [$\text{kJ} \cdot \text{kg}^{-1}$]

n_i^{VSP} - molární množství vlhkých spalin [mol]

n_i^P - molární množství paliva [mol]

h_i^P - měrné entalpie paliva [$\text{kJ} \cdot \text{mol}^{-1}$]

h_i^{VSP} - měrné entalpie vlhkých spalin [$\text{kJ} \cdot \text{mol}^{-1}$]

- **Adiabatická teplota plamene**

Určení teploty plamene při adiabatickém spalování vychází z předpokladu, že neexistují tepelné ztráty, a teplo vytvořené spálením reaktantu (paliva + okysličovadla) zvýší entalpii spalin a zároveň jejich teplotu. Vycházíme proto z rovnice [105]:

$$\sum_{i=1}^n n_i^{VSP} \cdot h_i^{VSP} = \sum_{i=1}^n n_i^P \cdot h_i^P + \sum_{i=1}^n n_i^{OK} \cdot h_i^{OK} + lhv \quad 7.5.18$$

Kde:

n_i^{OK} - molární množství okysličovadla [mol]

h_i^{OK} - měrná entalpie okysličovadla [$\text{kJ} \cdot \text{mol}^{-1}$]

Protože známe hodnotu na pravé straně rov. 7.5.18 a z rovnice 7.5.13 víme, že entalpie jsou funkcí teploty, lze iterací teploty najít entalpie vlhkých spalin, pro které bude rovnice platit.

7.5.3. Program:

Použité knihovny:

- Cantera;
- Numpy.

Použité nástroje:

- Jupyter Notebook.

Kód je součástí Jupyter notebooku a je rozdělený do 7 buněk. Kromě první buňky, kde je definované zadání, jsou buňky rozděleny stejným způsobem jako uvedená teorie. Teorie ale slouží pro seznámení s problematikou výpočtu. Samotný výpočet pomocí programu se od uvedené teorie liší v následujícím:

- Výpočet, kolik je potřeba množství kyslíku na spálení paliva, je řešen pomocí funkce `get_equivalence_ratio` modulu `Quantity`.
- Výpočet složení spalin je automaticky řešený pomocí funkce `equilibrate` (modul `Solution`) za konstantního tlaku a teploty.

- V programu není entalpie počítána přímo dle rov. 7.5.13. Tento výpočet počítá Cantera na pozadí a uživatel pouze odečte hodnoty entalpií z modulu *Solution* (definuje složení dané směsi) pomocí funkce *enthalpy_mole* (J/kmol) nebo *enthalpy_mass* (J/kg). Hodnoty entalpie jsou vztaženy k teplotě 273,15 K.
- Hodnota adiabatického plamene je získána odečtením teploty a použitím funkce *equilibrate* (modul *Solution*) za konstantní teploty a entalpie na směs okysličovadla a paliva.

Výše uvedené funkce jsou řešeny za použití knihovny Cantera. Základní soubor gri30.cti, který je součástí Cantery, neobsahuje vyšší uhlovodíky, pro výpočet byl použit GRIMECH soubor *JetSurf*. Na několika prvcích (CH₄, CO₂, N₂) bylo zkoušeno, jestli se hodnoty entalpií vzájemně neliší, ale protože byly identické a oba soubory vycházejí z NASA 7 polynomů, tak tomu autor důvěřuje.

V programu je použita i knihovna Numpy, ale její použití není klíčové a je využita jen pro přidání vodné páry a jejího množství do n-tic, které definují složení suchého vzduchu. Šlo to udělat čistě pomocí Pythonu, ale díky Numpy to bylo jednodušší.

7.5.4. Kontrola výpočtu:

Zadání bylo nastaveno tak, aby výsledky šly porovnat s výsledky [106].

Výsledky jsou totožné (viz. 7.3), kromě dvou věcí:

1. Parciální tlak nasycené vodní páry – nepatrně rozdílná hodnota p.t.n.v.p. (Burngas 2329,56 Pa a Cantera 2333,62 Pa) se promítá do složení okysličovadla a s tím vzniká odchylka i u složení spalin a průtoku okysličovadla.
2. Entalpie spalin – v hodnotě entalpie paliva je odchylka kolem 5,9 %. Ta dle mého názoru není způsobena chybným výpočtem, jelikož stejný postup byl použit pro výpočet entalpie okysličovadla a spalin a u obou je rozdíl menší než 1 %, ale vznikla v důsledku použití rozdílného zdroje dat pro výpočet entalpií. V programu Burngas se vychází z [109] a z hodnot cp používaných na ÚPEI FSI VUT Brno.

Tab. 7.3 Porovnání hodnot – Spalování plynné směsi

		Burngas	Cantera výpočet	Odchylka od Burngas [%]
Složení okysličovadla [%]	N2	77,140	77,141	-0,001296344
	O2	20,748	20,748	0
	Ar	0,929	0,929	0
	CO2	0,030	0,030	0
	H2O	1,153	1,152	0,086730269
Složení spalin [%]				
Vlhké	N2	70,067	70,068	-0,001427205
	Ar	0,843	0,843	0
	CO2	9,915	9,915	0
	H2O	19,175	19,174	0,005215124
Suché	N2	86,690	86,690	0
	Ar	1,043	1,043	0
	CO2	12,267	12,267	0
Průtoky [m3/hod]	Okysličovadla	15859,290	15859,019	0,001708778
	Vlhkých spalin	17480,715	17480,444	0,00155028
	Suchých Spalin	14128,769	14128,769	0
Výhřevnost (LHV) [MJ/kg]		46,853	46,886	-0,070983303
Teplota spalin [K]		2313,150	2311,192	0,084646478
Entalpie [kJ/kmol]	Paliva	805,064	757,459	5,913194479
	Okysličovadla	581,309	581,354	-0,00774115
	Spalin	76374,335	76442,736	-0,089560191

8. Závěr

I přes snahu zahrnout do výuky na školách programování, velká část lidí (mimo IT oblast) stále používá pro výpočty Excel. Tato práce nevyvrátila na první pohled neotřesitelnou pozici, jakou Excel má v řadě firem, ale spíše si dala za cíl ukázat, kde všude lze Python použít, a vyvrátit společenský mýtus, že programování musí být složité.

Práce byla rozdělena na teoretickou a praktickou část. Teoretická část byla rozdělena do pěti kapitol, kde první dvě ukázaly, proč je Python tak oblíbený mezi vědeckou komunitou, porovnaly využití s již zmíněným Excelem a rozšířily povědomí o možnostech využití Pythonu (z pohledu strojírenské praxe). Následně byly představeny základní studijní zdroje a ukázka jednoduchého nastavení Pythonu z pohledu uživatele, který má minimální zkušenosti s programováním obecně. Jelikož programování nejenom Pythonu se v dnešní době točí okolo specializovaných knihoven, je jim věnována poslední teoretická kapitola. Výběr knihoven je kombinací oblíbených vědeckých knihoven (Scipy, Numpy, Matplotlib, Pandas), materiálových knihoven (Cantera, CoolProp, IAPWS) a několika doplňkových knihoven (Openpyxl, PyQt a Jupyter Widgets).

Praktická část práce ukazuje použití Pythonu na pěti příkladech z oblasti termomechaniky a hydromechaniky, u kterých bylo demonstrováno použití již zmíněných knihoven. Veškeré příklady (krom jednoho), byly zpracovány v Jupyter Notebooku.

Z práce vyplývá, že v případě potřeby lze Python použít jako náhradu či jako doplněk k Excelu (knihovny Pandas a Openpyxl). Díky existenci již zmíněných materiálových knihoven se lze relativně jednoduše a bezplatně dostat k široké škále referenčních dat velkého množství plynů a kapalin. Kromě toho práce nastínila i možné využití v rámci automatizace řady běžně používaných softwarů (např. ANSYS) a možnost propojení s hardwarem (např. Arduino).

Ze zkušeností autora práce se základy Pythonu dají naučit za pár hodin, a pomineme-li knihovnu PyQt, která vyžaduje pokročilejší znalosti programování (základy OOP), je použití ostatních knihoven podobně intuitivní.

9. Seznam použité literatury

- [1] *Presidential Memorandum for the Secretary of Education | The White House* [online]. [vid. 2019-05-06]. Dostupné z: <https://www.whitehouse.gov/presidential-actions/presidential-memorandum-secretary-education/>
- [2] *American companies fund \$300M in largest joint effort for K-12 computer science* [online]. [vid. 2019-05-06]. Dostupné z: <https://medium.com/@codeorg/american-companies-fund-300m-in-largest-joint-effort-for-k-12-computer-science-763a33639818>
- [3] *Strategie digitálního vzdělávání do roku 2020, MŠMT ČR* [online]. [vid. 2019-05-06]. Dostupné z: <http://www.msmt.cz/vzdelavani/skolstvi-v-cr/strategie-digitalniho-vzdelavani-do-roku-2020>
- [4] *The History of Python: A Brief Timeline of Python* [online]. [vid. 2019-05-06]. Dostupné z: <http://python-history.blogspot.com/2009/01/brief-timeline-of-python.html>
- [5] *GitHub lists the 15 most popular programming languages - Business Insider* [online]. [vid. 2019-05-06]. Dostupné z: <https://www.businessinsider.com/the-9-most-popular-programming-languages-according-to-the-facebook-for-programmers-2017-10>
- [6] FERRARI, A. *python introduzione e primi programmi* [online]. nedatováno [vid. 2019-05-16]. Dostupné z: <http://www.ferrarialberto.com/scuola/as1819/3/lezioni/c3-02-python-introduzione.pdf>
- [7] *API | computer programming | Britannica.com* [online]. [vid. 2019-05-06]. Dostupné z: <https://www.britannica.com/technology/API>
- [8] *Gmail API | Google Developers* [online]. [vid. 2019-05-06]. Dostupné z: <https://developers.google.com/gmail/api/>
- [9] *FreeCAD: FreeCAD source documentation* [online]. [vid. 2019-05-06]. Dostupné z: <https://www.freecadweb.org/api/>
- [10] DESAI, P. *Python programming for Arduino*. B.m.: Packt Publishing Ltd, 2015.
- [11] *Raspberry Pi — Teach, Learn, and Make with Raspberry Pi* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.raspberrypi.org/>
- [12] *The MicroPython project* [online]. Dostupné z: <https://github.com/micropython/micropython>
- [13] *Say „Hello World” in 28 Different Programming Languages - Excel with Business* [online]. [vid. 2019-05-06]. Dostupné z: <https://excelwithbusiness.com/blog/say-hello-world-in-28-different-programming-languages/>
- [14] *Computer program | Britannica.com* [online]. [vid. 2019-05-06]. Dostupné z: <https://www.britannica.com/technology/computer-program#ref65437>
- [15] *PythonSpeed - Python Wiki* [online]. [vid. 2019-05-06]. Dostupné z: <https://wiki.python.org/moin/PythonSpeed>
- [16] *A Speed Comparison Of C, Julia, Python, Numba, and Cython on LU Factorization (IT Best Kept Secret Is Optimization)* [online]. [vid. 2019-05-06]. Dostupné z: https://www.ibm.com/developerworks/community/blogs/jfp/entry/A_Comparison_O

- f_C_Julia_Python_Numba_Cython_Scipy_and_BLAS_on_LU_Factorization?lang=en
- [17] *The 2018 Top Programming Languages - IEEE Spectrum* [online]. [vid. 2019-05-06]. Dostupné z: <https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>
 - [18] *Most Popular and Influential Programming Languages of 2018* [online]. [vid. 2019-05-06]. Dostupné z: <https://stackify.com/popular-programming-languages-2018/>
 - [19] *Stack Overflow - Where Developers Learn, Share, & Build Careers* [online]. [vid. 2019-05-10]. Dostupné z: <https://stackoverflow.com/>
 - [20] ROBINSON, D. *Why is Python Growing So Quickly?* [online]. Dostupné z: <https://stackoverflow.blog/2017/09/14/python-growing-quickly/>
 - [21] *Choosing a License — The Hitchhiker's Guide to Python* [online]. [vid. 2019-05-06]. Dostupné z: <https://docs.python-guide.org/writing/license/>
 - [22] *Licenses by Name | Open Source Initiative* [online]. [vid. 2019-05-06]. Dostupné z: <https://opensource.org/licenses/alphabetical>
 - [23] *Technické údaje a omezení aplikace Excel - Excel* [online]. [vid. 2019-05-10]. Dostupné z: <https://support.office.com/cs-cz/article/technické-údaje-a-omezení-aplikace-excel-1672b34d-7043-467e-8e27-269d656771c3>
 - [24] *Mistakes in Excel: The 5 Biggest Spreadsheet Fails | Professor Excel | Professor Excel* [online]. [vid. 2019-05-10]. Dostupné z: <https://professor-excel.com/mistakes-in-excel-the-5-biggest-spreadsheet-fails/>
 - [25] *Excel errors: How Microsoft's spreadsheet may be hazardous to your health | ZDNet* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.zdnet.com/article/excel-errors-microsofts-spreadsheet-may-be-hazardous-to-your-health/>
 - [26] *How does Python compare to other solutions?* [online]. [vid. 2019-05-06]. Dostupné z: <http://scipy-lectures.org/intro/intro.html#why-python>
 - [27] *SchoolsUsingPython - Python Wiki* [online]. [vid. 2019-05-06]. Dostupné z: <https://wiki.python.org/moin/SchoolsUsingPython>
 - [28] *MAE-6226: Aerodynamics Course Syllabus* [online]. [vid. 2019-05-10]. Dostupné z: https://figshare.com/articles/MAE-6226_Aerodynamics_Course_Syllabus/4584328
 - [29] *Classical Aerodynamics with Python* [online]. Dostupné z: <https://github.com/barbagroup/AeroPython>
 - [30] *GW Open edX* [online]. [vid. 2019-05-06]. Dostupné z: <https://openedx.seas.gwu.edu/>
 - [31] *Using Computational Thinking to Teach Mechanical Vibrations – Engineering Education Learning Community* [online]. [vid. 2019-05-06]. Dostupné z: <https://engineering.ucdavis.edu/eelc/using-computational-thinking-to-teach-mechanical-vibrations/>
 - [32] *Resonance: Learning Mechanical Vibration Engineering Through Computation* [online]. Dostupné z: <https://github.com/moorepants/resonance>
 - [33] *CBE 20255: Intro to Chemical Engineering* [online]. [vid. 2019-05-06]. Dostupné z: <https://class->

search.nd.edu/reg/srch/ClassSearchServlet?CRN=11290&TERM=201510&P=208035

- [34] *Introduction to Chemical Engineering Analysis* [online]. Dostupné z: <http://jckantor.github.io/CBE20255/>
- [35] *Python scripting tutorial – FreeCAD Documentation* [online]. [vid. 2019-05-07]. Dostupné z: https://www.freecadweb.org/wiki/Python_scripting_tutorial
- [36] *Automated Dimensioning in FreeCAD - Python - YouTube* [online]. [vid. 2019-05-07]. Dostupné z: <https://www.youtube.com/watch?v=gOXH0m9HTII>
- [37] *Engineering Simulation & 3D Design Software / ANSYS* [online]. [vid. 2019-05-07]. Dostupné z: <https://www.ansys.com/>
- [38] Workbench Scripting Guide [online]. nedatováno [vid. 2019-05-07]. Dostupné z: https://itrss.mst.edu/media/informationtechnology/itrss/documents/ansysdocs/Workbench_Scripting_Guide.pdf
- [39] *How to use scripts to generate a plane wing using ANSYS Design Modeler 15.0 – Technical Content* [online]. [vid. 2019-05-06]. Dostupné z: <https://support.esss.co/hc/en-us/articles/205250985-How-to-use-scripts-to-generate-a-plane-wing-using-ANSYS-Design-Modeler-15-0>
- [40] *Abaqus Unified FEA - SIMULIA™ by Dassault Systèmes®* [online]. [vid. 2019-05-07]. Dostupné z: <https://www.3ds.com/products-services/simulia/products/abacus/>
- [41] *Abaqus Scripting User's Guide* [online]. Dostupné z: <https://www.sharcnet.ca/Software/Abaqus/6.14.2/v6.14/books/cmd/default.htm?startat=pt03ch07s01.html>
- [42] GASSARA, F., HAMBLI, R., BOURAOU, T., EL HALOUANI, F., SOULAT, D. Optimization of springback in L-bending process using a coupled Abaqus/Python algorithm. *Article in International Journal of Advanced Manufacturing Technology* [online]. 2009 [vid. 2019-05-07]. Dostupné z: doi:10.1007/s00170-008-1819-4
- [43] *Arduino* [online]. Dostupné z: <https://www.arduino.cc/>
- [44] *pySerial* [online]. Dostupné z: <https://github.com/pyserial/pyserial>
- [45] *Raspberry Pi — Teach, Learn, and Make with Raspberry Pi* [online]. [vid. 2019-05-07]. Dostupné z: <https://www.raspberrypi.org/>
- [46] SURESH, N., et al. RASPBERRY PI BASED LIQUID FLOW MONITORING AND CONTROL. *International Journal of Research in Engineering and Technology* [online]. 2014, 03(07), 122–125 [vid. 2019-05-07]. ISSN 23217308. Dostupné z: doi:10.15623/ijret.2014.0307020
- [47] *Knihovnička python.cz* [online]. [vid. 2019-05-07]. Dostupné z: <https://books.pyvo.cz/>
- [48] *Free Ebook Foundation promotes access and preservation of knowledge, literature, and culture.* [online]. [vid. 2019-05-07]. Dostupné z: <https://ebookfoundation.org/>
- [49] *MOOC / Definition of MOOC in English by Oxford Dictionaries* [online]. [vid. 2019-05-10]. Dostupné z: <https://en.oxforddictionaries.com/definition/mooc>
- [50] *What is a MOOC? - MOOC (Massive Open Online Course)* [online]. [vid. 2019-05-10]. Dostupné z: <http://desarrolloweb.dlsi.ua.es/moocs/what-is-a-mooc>

- [51] *edX / Free Online Courses by Harvard, MIT, & more* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.edx.org/>
- [52] *Coursera / Online Courses & Credentials by Top Educators. Join for Free* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.coursera.org/>
- [53] *Nanodegree Programs and Free Online Classes / Udacity* [online]. [vid. 2019-05-10]. Dostupné z: <https://eu.udacity.com/>
- [54] *A Comprehensive List of MOOC (Massive Open Online Courses) Providers* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.technoduet.com/a-comprehensive-list-of-mooc-massive-open-online-courses-providers/>
- [55] *List of 41 providers offering MOOCs/free online courses / Class Central* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.classcentral.com/providers>
- [56] *MOOC Platform Comparison Table / MoocLab - Connecting People to Online Learning* [online]. [vid. 2019-05-10]. Dostupné z: https://www.mooclab.club/pages/mooc_comparison_2018/
- [57] *Search MOOCs and Free Online Courses / MOOC List* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.mooc-list.com/tags/search>
- [58] *Class Central • #1 Search Engine for Free Online Courses & MOOCs* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.classcentral.com/>
- [59] ATWOOD, J. I wish more people understood ... [Twitter] [online]. 2008. Dostupné z: https://twitter.com/codinghorror/status/991082088689381376?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwterm%5E991082088689381376&ref_url=https%3A%2F%2Fblog.codinghorror.com%2F
- [60] *Google Groups: Get Started / Learning Center / G Suite* [online]. [vid. 2019-05-10]. Dostupné z: <https://gsuite.google.com/learning-center/products/groups/get-started/#/>
- [61] *Conda — Conda documentation* [online]. [vid. 2019-05-10]. Dostupné z: <https://conda.io/en/latest/>
- [62] *Conda: Myths and Misconceptions / Pythonic Perambulations* [online]. [vid. 2019-05-10]. Dostupné z: <http://jakevdp.github.io/blog/2016/08/25/conda-myths-and-misconceptions/>
- [63] *Welcome to Python.org* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.python.org/>
- [64] *Anaconda* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.anaconda.com/>
- [65] *Spyder Website* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.spyder-ide.org/>
- [66] *Spyder: The Scientific Python Development Environment — Documentation — Spyder 3 documentation* [online]. [vid. 2019-05-10]. Dostupné z: <https://docs.spyder-ide.org/index.html>
- [67] *Wolfram Mathematica: What's New in Mathematica 6* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.wolfram.com/mathematica/newin6/>
- [68] *PTC Mathcad / PTC* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.ptc.com/en/products/mathcad>

- [69] *Jupyterlab variableinspector* [online]. Dostupné z: <https://github.com/lckr/jupyterlab-variableInspector>
- [70] *Project Jupyter* [online]. [vid. 2019-05-10]. Dostupné z: <https://jupyter.org/>
- [71] *Top 5 Python IDEs For Data Science (article) - DataCamp* [online]. [vid. 2019-05-10]. Dostupné z: https://www.datacamp.com/community/tutorials/data-science-python-ide?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adposition=1t2&utm_creative=332602034364&utm_targeti
- [72] *Vertabelo Academy Blog / 5 Best Python IDEs for Data Science* [online]. [vid. 2019-05-10]. Dostupné z: <https://academy.vertabelo.com/blog/5-best-python-ide-data-science/>
- [73] *NumPy — NumPy* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.numpy.org/>
- [74] *Nauč se Python! - NumPy* [online]. [vid. 2019-05-10]. Dostupné z: <https://naucese.python.cz/lessons/intro/numpy/>
- [75] *SciPy.org — SciPy.org* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.scipy.org/>
- [76] *Vědecká knihovna SciPy | Vědecké programování v Pythonu* [online]. [vid. 2019-05-10]. Dostupné z: <https://pythonic.eu/fjfi/posts/vedecka-knihovna-scipy.html>
- [77] HUNTER, J. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* [online]. 2007, 9(3), 90–95 [vid. 2019-05-10]. ISSN 1521-9615. Dostupné z: doi:10.1109/MCSE.2007.55
- [78] *Matplotlib - grafy v Pythonu | Vědecké programování v Pythonu* [online]. [vid. 2019-05-21]. Dostupné z: <https://pythonic.eu/fjfi/posts/matplotlib.html>
- [79] *Python Data Analysis Library — pandas: Python Data Analysis Library* [online]. [vid. 2019-05-10]. Dostupné z: <https://pandas.pydata.org/index.html>
- [80] *ipywidgets — Jupyter Widgets 7.4.2 documentation* [online]. [vid. 2019-05-10]. Dostupné z: <https://ipywidgets.readthedocs.io/en/stable/index.html>
- [81] *Project Jupyter – Skupiny Google* [online]. [vid. 2019-05-10]. Dostupné z: <https://groups.google.com/forum/#!forum/jupyter>
- [82] *Riverbank / Software / PyQt / What is PyQt?* [online]. [vid. 2019-05-10]. Dostupné z: <https://www.riverbankcomputing.com/software/pyqt/intro>
- [83] FOROGH, P. *PyQt5 Introduction And First Window Class Python GUI Programming* [online]. 2017. Dostupné z: <https://www.youtube.com/watch?v=pnpL9Sl79g8&list=PL1FgJUcJJ03uwFW8ys2oV2dffKs3ieGYk>
- [84] *Qt Designer Manual* [online]. [vid. 2019-05-10]. Dostupné z: <https://doc.qt.io/qt-5/qtdesigner-manual.html>
- [85] SUMMERFIELD, M. *Rapid GUI programming with Python and Qt: the definitive guide to PyQt programming*. B.m.: Pearson Education., 2007.
- [86] *iapws* [online]. Dostupné z: <https://github.com/jjgomera/iapws>
- [87] *Welcome to iapws's documentation! — iapws 1.1.3 documentation* [online]. [vid. 2019-

- 05-10]. Dostupné z: <https://iapws.readthedocs.io/en/latest/>
- [88] BELL, I. H., WRONSKI, J., QUOILIN, S., & LEMORT, V. Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp. *Industrial & Engineering Chemistry Research* [online]. 2014, **53**(6), 2498–2508 [vid. 2019-05-10]. ISSN 0888-5885. Dostupné z: doi:10.1021/ie4033999
 - [89] *Coolprop-users – Skupiny Google* [online]. [vid. 2019-05-10]. Dostupné z: <https://groups.google.com/forum/#!forum/coolprop-users>
 - [90] GOODWIN, D. G., MOFFAT, H. K., & SPETH, R. L. *Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes* [online]. 2018. Dostupné z: <https://www.cantera.org>
 - [91] SHEPHERD, J. *Chemical Reaction and Thermodynamic Data - Cantera Format* [online]. Dostupné z: http://shepherd.caltech.edu/EDL/PublicResources/sdt/cti_mech.html
 - [92] FELDEN, A. *CANTERA Tutorials A series of tutorials to get started with the python interface of Cantera version 2.1.1* [online]. 2015 [vid. 2019-05-10]. Dostupné z: <http://cantera.github.io/docs/sphinx/html/index.html>
 - [93] *Cantera Users' Group – Skupiny Google* [online]. [vid. 2019-05-10]. Dostupné z: <https://groups.google.com/forum/#!forum/cantera-users>
 - [94] *openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files — openpyxl 2.6.2 documentation* [online]. [vid. 2019-05-10]. Dostupné z: <https://openpyxl.readthedocs.io/en/stable/>
 - [95] ŠOB, F. *Hydromechanika*. B.m.: Brno, CERN, 2001.
 - [96] PAVELEK, M. a kol. *Termomechanika. FSI VUT v Brně*. 2011.
 - [97] *Termomechanika* [online]. [vid. 2019-05-09]. Dostupné z: <http://home.zcu.cz/~kovarikp/termomechanika.html>
 - [98] *e-zatepleni.cz | Tepelné izolace | Fasádní polystyren, omítky, štuky a stěrky* [online]. [vid. 2019-05-09]. Dostupné z: <https://www.e-zatepleni.cz/>
 - [99] BERGMAN, T. L., INCROPERA, F. P., DEWITT, D. P., & LAVINE, A. S. *Fundamentals of heat and mass transfer*. SIXTH EDIT. B.m.: Wiley, 2007. ISBN 0-471-45728-0.
 - [100] KADLEC, Z. *Termomechanika návody na cvičení*. B.m.: VŠB-TUO Ostrava, 2012. ISBN 978-80-248-1736-1.
 - [101] *Hodnoty fyzikálních veličin vybraných stavebních materiálů* [online]. Dostupné z: <https://stavba.tzb-info.cz/tabulky-a-vypocty/58-hodnoty-fyzikalnich-velicin-vybranych-stavebnich-materialu>
 - [102] FELDER, R. M., ROUSSEAU, R. W., & BULLARD, L. G. *Elementary Principles of Chemical Processes*. 3rd Update. B.m.: Wiley, 2004. ISBN 0-471-68757-X.
 - [103] THOL, Monika, Roland SPAN a Eric W LEMMON. *Equation of state for benzene for temperatures from the melting line up to 725 K with pressures up to 500 MP Speed of Sound Measurements View project Equation of state for benzene for temperatures from*

- the melting line up to 725 K with pressures up to 500 MPa* † [online]. nedatováno [vid. 2019-05-19]. Dostupné z: <https://www.researchgate.net/publication/269275284>
- [104] LEMMON, Eric W. a Roland SPAN. Short Fundamental Equations of State for 20 Industrial Fluids. *Journal of Chemical & Engineering Data* [online]. 2006, **51**(3), 785–850 [vid. 2019-05-19]. ISSN 0021-9568. Dostupné z: doi:10.1021/je050186n
- [105] CENGEL, Y. A., BOLES, M. A. *Thermodynamics in engineering approach*. EIGHTH EDI. B.m.: Mc Graw-Hill, 2015. ISBN 978-0-07-339817-4.
- [106] NEJEZCHLEB, R. Fyzikální vlastnosti a spalovací charakteristiky paliv. *Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství*. 2009, 35.
- [107] *Cantera* [online]. [vid. 2019-05-09]. Dostupné z: <https://cantera.org/index.html>
- [108] MCBRIDE, B. J., ZEHE, M. J., & GORDON, S. NASA Glenn coefficients for calculating thermodynamic properties of individual species [online]. 2002. Dostupné z: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20020085330.pdf>
- [109] STEHLÍK, P. *Termofyzikální vlastnosti. Tepelné pochody: Teoretické základy oboru*. B.m.: Brno: VUT Brno, 1992. ISBN 80-214-0428-0.

10. Seznam použitých zkratek a symbolů

Symbols

A_1	exp. získané hodnoty pro konkrétní prvek	[-]
A_2	exp. získané hodnoty pro konkrétní prvek	[-]
A_3	exp. získané hodnoty pro konkrétní prvek	[-]
A_4	exp. získané hodnoty pro konkrétní prvek	[-]
A_5	exp. získané hodnoty pro konkrétní prvek	[-]
A_6	exp. získané hodnoty pro konkrétní prvek	[-]
AFR	poměr mezi množstvím okysličovadla a paliva	[-]
C_{izol}	cena materiálu izolace	[Kč]
C_{plyn}	cena plynu	[Kč·kWh ⁻¹]
C_{plyn}	cena plynu	[Kč·kWh ⁻¹]
$C_{wH_2O}^{VSP}$	hm. koncentrace vody ve vlhkých spalínách	[%]
c_{ni}	molární/objemová koncentrace	[%]
$c_{nO_2}^{OK}$	molární koncentrace kyslíku v okysličovadle	[%]
c_{wi}	hmotnostní koncentrace	[%]
g	gravitační zrychlení	[m·s ⁻²]
h	výška hladiny	[m]
h_1	měrná entalpie kapaliny	[kJ·kg ⁻¹]
h_2	měrná entalpie syté kapaliny	[kJ·kg ⁻¹]
$h_{298.15-0}$	měrné ent. pro 298.15 K vztažená k 0 K	[kJ·mol ⁻¹]
h_3	měrná entalpie syté páry	[kJ·kg ⁻¹]
h_{T-0}	měrná entalpie vztažená k 0 K	[kJ·mol ⁻¹]
h_i^{OK}	měrná entalpie okysličovadla	[kJ·mol ⁻¹]
h_i^P	měrné entalpie paliva	[kJ·mol ⁻¹]
h_i^{VSP}	měrné entalpie vlhkých spalín	[kJ·mol ⁻¹]
lhv	výhřevnost	[kJ·mol ⁻¹]
LHV	výhřevnost	[kJ·kg ⁻¹]

l_{BET}	tloušťka stěny	[m]
l_{IZOL}	tloušťka izolace	[m]
l_{PER}	tloušťka perlitové omítky	[m]
l_{VAP}	tloušťka vápenné omítky	[m]
M	molární hmotnost	[kg/kmol]
m	hmotnost	[kg]
\dot{m}	hmotnostní tok	[kg·s ⁻¹]
m'	hmotnost syté kapaliny	[kg]
m''	hmotnost syté páry	[kg]
m_i	hmotnost dané složky	[kg]
\dot{m}^{OK}	hmotnostní tok okysličovadla	[kg·hod ⁻¹]
\dot{m}^P	hmotnostní tok paliva	[kg·hod ⁻¹]
\dot{m}^{SP}	hmotnostní tok suchých spalin	[kg·hod ⁻¹]
\dot{m}^{VSP}	hmotnostní tok vlhkých spalin	[kg·hod ⁻¹]
n	molární množství	[kmol]
\dot{n}_{O_2}	molový tok kyslíku	[kmol·hod ⁻¹]
\dot{n}^{OK}	molový tok okysličovadla	[kmol·hod ⁻¹]
\dot{n}^P	molový tok paliva	[kmol·hod ⁻¹]
n_i^{OK}	molární množství okysličovadla	[mol]
n_i^P	molární množství paliva	[mol]
n_i^{VSP}	molární množství vlhkých spalin	[mol]
n_i	molární množství dané složky	[kmol]
N_V	molární objem = 22,414	[m ³ ·kmol ⁻¹]
P	výkon	[kW]
p	tlak	[Pa]
p_i	parciální tlak dané složky	[Pa]
Q	průtok	[m ³ ·s ⁻¹]
\dot{Q}	tepelný tok	[W]
\dot{Q}_{izol}	tepelné ztráty izolovanou stěnou	[kW]
\dot{Q}_{neizol}	tepelné ztráty neizolovanou stěnou	[kW]

q	měrné celkové teplo	$[\text{kJ} \cdot \text{kg}^{-1}]$
q_{12}	měrné teplo	$[\text{kJ} \cdot \text{kg}^{-1}]$
q_{23}	měrné teplo	$[\text{kJ} \cdot \text{kg}^{-1}]$
R	tepelný odpor	$[\text{K} \cdot \text{W}^{-1}]$
R_m	univerzální plynová konstanta = 8,3143	$[\text{J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}]$
R_{izol}	tepelný odpor izolované stěny	$[\text{K} \cdot \text{W}^{-1}]$
R_{neizol}	tepelný odpor neizolované stěny	$[\text{K} \cdot \text{W}^{-1}]$
r	měrná plynová konstanta	$[\text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}]$
S	plocha stěny	$[\text{m}^2]$
S_1	průřez nádrže	$[\text{m}^2]$
S_2	průřez výtoku	$[\text{m}^2]$
S_{iO_2}	stechiometrický koeficient udávající množství kyslíku	$[-]$
s_2	měrná entropie syté kapaliny	$[\text{kJ} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}]$
s_3	měrná entropie syté páry	$[\text{kJ} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}]$
s_x	měrná entropie mokré páry	$[\text{kJ} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}]$
t	čas	$[\text{s}]$
t_{inv}	návratnost investice	$[\text{hod}]$
T	teplota	$[\text{K}]$
T_{in}	průměrná teplota uvnitř budovy	$[\text{°C}]$
T_{out}	průměrná venkovní teplota	$[\text{°C}]$
V	objem	$[\text{m}^3]$
V_i	objem dané složky	$[\text{m}^3]$
v	měrný objem	$[\text{m}^3/\text{kg}]$
v_{kap}	rychlost kapaliny	$[\text{m} \cdot \text{s}^{-1}]$
v_2	měrný objem syté kapaliny	$[\text{m}^3 \cdot \text{kg}^{-1}]$
v_3	měrný objem syté páry	$[\text{m}^3 \cdot \text{kg}^{-1}]$
v_x	měrný objem mokré páry	$[\text{m}^3 \cdot \text{kg}^{-1}]$
w_i	hmotnostní zlomek	$[-]$
x_i	molární/objemový zlomek	$[-]$
$Y_{z\ 1,2}$	měrná ztrátová energie	$[\text{J} \cdot \text{kg}^{-1}]$

z	tlaková výška	[m]
α	součinitel přestupu tepla	[W·m ⁻² ·K ⁻¹]
α_{in}	součinitel přestupu tepla na vnitřní straně stěny	[W·m ⁻² ·K ⁻¹]
α_{in}	součinitel přestupu tepla na vnitřní straně stěny	[W·m ⁻² ·K ⁻¹]
α_{out}	součinitel přestupu tepla na vnější straně	[W·m ⁻² ·K ⁻¹]
α_{out}	součinitel přestupu tepla na vnější straně stěny	[W·m ⁻² ·K ⁻¹]
ΔT	rozdíl teplot	[K]
δ	šířka stěny	[m]
λ	tepelná vodivost	[W·m ⁻¹ ·K ⁻¹]
λ_{BET}	tepelná vodivost železobetonu	[W·m ⁻¹ ·K ⁻¹]
λ_{PER}	tepelná vodivost perlitové omítky	[W·m ⁻¹ ·K ⁻¹]
λ_{VAP}	tepelná vodivost vápenné omítky	[W·m ⁻¹ ·K ⁻¹]
ρ	hustota	[kg·m ⁻³]
φ	rychlostní součinitel	[-]

Zkratky:

API	Rozhraní pro programování aplikací (Application Programming Interface)
GUI	Uživatelské prostředí (Graphical User Interface)
IDE	Vývojové prostředí (Integrated Development Environment)
MOOC	Online kurz s neomezeným množstvím uživatelů (Massive Open Online Course)
OOP	Objektově orientované programování (Object-oriented programming)

11. Seznam příloh

- Příloha 1 - Seznam instalovaných knihoven;
- Příloha 2 - Návod k programu Výparník;

Příloha 1

Seznam instalovaných knihoven:

Pandas – ver. 0.23.4

```
conda install -c anakonda pandas=0.23.4
```

Cantera – ver. 2.4.0

```
conda install -c cantera cantera=2.4.0
```

CoolProp – ver. 6.2.1

```
conda install -c conda-forge coolprop=6.2.1
```

IAPWS – ver. 1.3

```
conda install -c conda-forge iapws=1.3
```

ipywidgets – ver. 7.4.2

```
conda install -c anaconda ipywidgets=7.4.2
```

Matplotlib – ver. 3.0.1

```
conda install -c conda-forge matplotlib=3.0.1
```

Numpy

není nutné instalovat

openpyxl

```
conda install -c anaconda openpyxl=2.5.11
```

pyqt

není nutné instalovat

qt

není nutné instalovat

scipy

není nutné instalovat

xlrd

```
conda install -c anaconda xlrd
```

lxml

```
conda install -c anaconda lxml
```

U knihoven, kde není uveden instalační odkaz tak byly součástí instalace některé z uvedených knihoven.

Příloha 2

Návod k programu Výparník

Program se ovládá následujícím způsobem:

1. Nejprve se musí zadat hodnoty tlaku a počáteční teploty do kolonky *Vstupní hodnoty* (viz. Obr.1 - příloha).

The screenshot shows a software window titled "Výparník". It features a "Menu" section at the top. The main interface is split into two panels. The left panel, labeled "Vstupní hodnoty", contains three input fields for "Počáteční teplota [°C]", "Tlak [kPa]", and "Hmotnostní tok [kg/s]". The right panel, labeled "Výstupní hodnoty", contains four output fields for "Koncová teplota [°C]", "Měrné latentní teplo [kJ/kg]", "Měrné celkové teplo [kJ/kg]", and "Výkon [kW]". Below these panels are two buttons: "Vypočítat" and "Uložit data". A large yellow rectangular area is positioned at the bottom of the window.

Obr.1- příloha Uživatelské prostředí výpočtu výparníku

2. Na rozdíl od uživatelského prostředí v kap. 7.1, kde byl výpočet automatický, je zde nutné zmáčknout tlačítko *Vypočítat*. Než se zobrazí hodnoty v kolonce *Výstupní hodnoty*, program zadané hodnoty zkontroluje. Kontroluje se:

- Jestli uživatel zadal číselnou hodnotu. Pokud uživatel zadá text, program ho upozorní hláškou

Vložte číselnou hodnotu

Tato hláška se zobrazí v textovém poli pod tlačítky *Vypočítat* a *Uložit data*. Pozor i na oddělování desetinných míst. Program desetinnou čárku vyhodnocuje, jako kdyby uživatel zadal text.

- Pokud jsou zadané hodnoty pod trojným bodem nebo nad kritickým bodem, program vypíše hlášku:

*Zvolte menší (větší) teplotu (tlak), než je ****

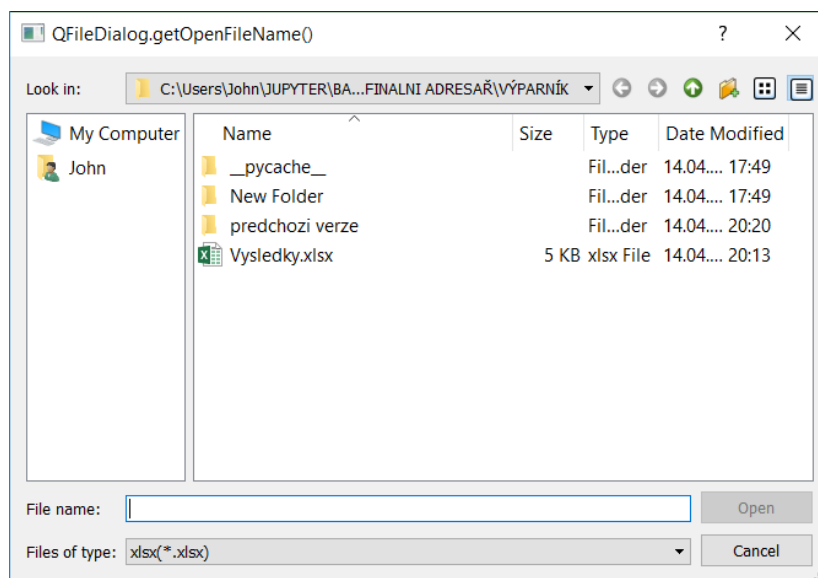
Pokud jsou obě hodnoty mimo již zmíněné body, program nejprve upozorní na změnu teploty a při dalším zmáčknutí tlačítka *Vypočítat* zobrazí upozornění pro změnu tlaku.

- Pokud jsou hodnoty v oblasti suché páry, program upozorní uživatele hláškou:

*Jste v oblasti suché páry. Zvyšte tlak nad *** kPa nebo snižte teplotu pod *** °C*

Pokud hodnoty prošly kontrolou, zobrazí se výsledky v kolonce *Výstupní hodnoty*.

3. Vypočtené hodnoty lze uložit do excelového souboru. Výběr souboru se řeší stisknutím tlačítka *Uložit data*, kterým vyvoláme dialogové okno pro výběr souboru. Je však nutné, aby daný soubor nebyl otevřený.



Obr. 2 – příloha Výběr souboru

Kromě složek se zobrazují pouze excelové soubory s koncovkou *xlsx*. Po stisknutí tlačítka *Open* se data uloží do vybraného souboru a v programu se zobrazí hláška:

Data uložena

V opačném případě se vypíše:

Nevybrán žádný soubor

V případě, kdy nejsou žádná data v kolonce *Výstupní hodnoty* a uživatel zvolí soubor, zobrazí se pouze hlášení:

Žádná data k uložení

4. Pro ukládání dat je součástí úlohy i excelový soubor, do kterého se hodnoty ukládají (viz. Obr.3 – příloha)

	A	B	C
1	Zadané hodnoty		
2	Počáteční teplota - t1	200,0000 °C	
3	Tlak - p	12000,0000 kPa	
4			
5	Vypočítané hodnoty		
6	Koncová teplota - t3	324,6780 °C	
7	Měrné latentní teplo - qlat	1194,2560 kJ/kg	
8	Měrné celkové teplo - q	1828,7950 kJ/kg	
9	Výkon - P	18287,9460 kW	
10			
11	Ostatní hodnoty		
12	Entalpie v bodě 1 - h1	856,7880 kJ/kg	
13	Entalpie v bodě 2 - h2	1491,3270 kJ/kg	
14	Entalpie v bodě 3 - h3	2685,5830 kJ/kg	
15	Měrný objem v bodě 1 - v1	0,001146 m3/kg	
16	Měrný objem v bodě 2 - v2	0,001526 m3/kg	
17	Měrný objem v bodě 3 - v3	0,014269 m3/kg	
18	Entropie v bodě 1 - s1	2,314673 kJ/kg*K	
19	Entropie v bodě 2 - s2	3,496459 kJ/kg*K	
20	Entropie v bodě 3 - s3	5,494116 kJ/kg*K	
21			

Obr.3 - příloha Uložená data

Ukládají se vždy pouze hodnoty z kolonky *Výstupní hodnoty*. Vstupní hodnoty, z nichž byl výpočet realizován, má program uložený v paměti, a jsou tedy nezávislé na aktuálních hodnotách v kolonce *Vstupní hodnoty*.